AD-A088 420 ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG--ETC F/G 1g/4
DEVELOPMENT OF A PROGRAM FOR COMPUTER-BASED RESISTANCE ANALYSIS--ETC(U)
JUL 80 J B CHEEK

UNCLASSIFIED WES/TR/SL-80-5 NL

1 OF 2
AD A088420

TECHNICAL REPORT SL-80-5

DEVELOPMENT OF A PROGRAM FOR COMPUTER-BASED RESISTANCE ANALYSIS (COBRA)

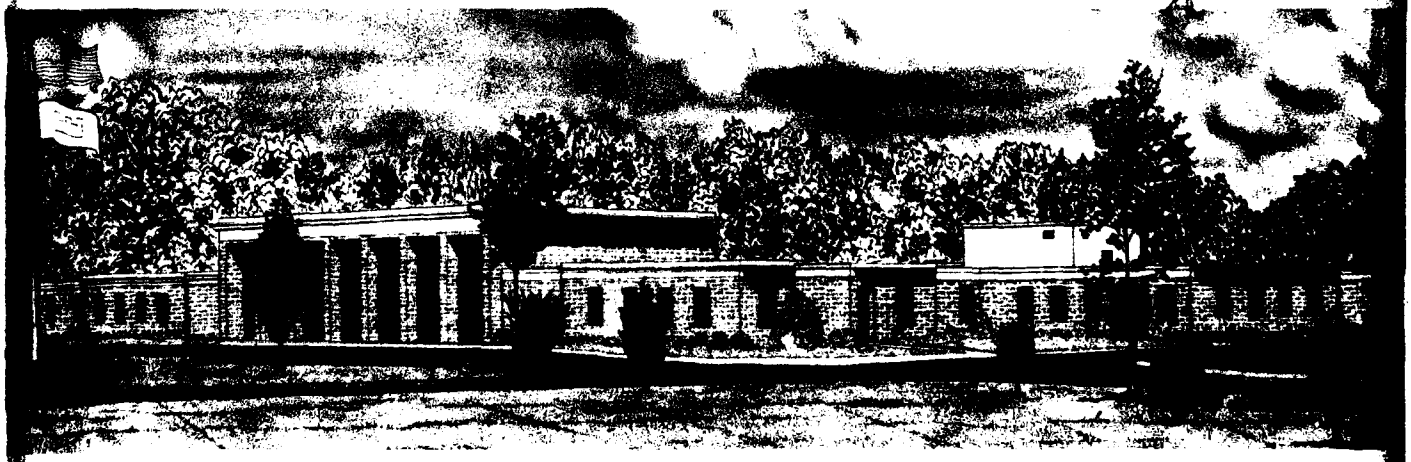
by

James B. Cheek, Jr.

Structures Laboratory
U. S. Army Engineer Waterways Experiment Station
P. O. Box 631, Vicksburg, Miss. 39180

July 1980

Final Report



Prepared for Office, Chief of Engineers, U. S. Army
Washington, D. C. 20314

Under Project No. 4A762719AT40, Task A1, Work Unit 003
and Task A4, Work Unit 001

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

(14) WES/TR/SL-80-5

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report SL-80-5	2. GOVT ACCESSION NO.: AD-A088 420
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): DEVELOPMENT OF A PROGRAM FOR COMPUTER-BASED RESISTANCE ANALYSIS (COBRA)	5. TYPE OF REPORT & PERIOD COVERED: Final report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): James B. Cheek, Jr.	8. CONTRACT OR GRANT NUMBER(s): (11) Jul
9. PERFORMING ORGANIZATION NAME AND ADDRESS: U. S. Army Engineer Waterways Experiment Station, Structures Laboratory, P. O. Box 631, Vicksburg, Miss. 39180	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project No. 4A762719AT40, Task A1, Work Unit 003, and Task A4, Work Unit 001
11. CONTROLLING OFFICE NAME AND ADDRESS: Office, Chief of Engineers, U. S. Army, Washington, D. C. 20314	12. REPORT DATE: July 1980
	13. NUMBER OF PAGES: 92
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

COBRA (Computer program)	Protective structures
Conventional weapons	Random processes
Hardened installations	Structural analysis
Projectiles	Weapons effects

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the development of a computer-based approach to evaluating the hardness of structures subjected to attack by nonnuclear weapons. The program has the following features:

a. It introduces a normal random change in projectile impact point with respect to aim point. This models the targeting uncertainty of weapon delivery systems.

b. It accounts for both blast- and fragment-induced damage

(continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

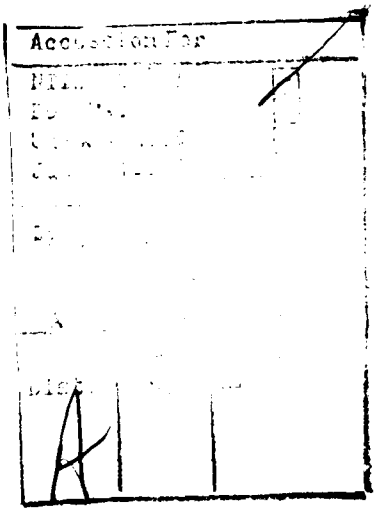
1 of 415

20. ABSTRACT (Continued)

for each shot.

c. It models the cumulative damage and the resultant increase in vulnerability produced by a series of shots (an attack).
d. It considers the redundancy of structural elements and approximately models their structural behavior with respect to blast-induced failure.
e. It models the curved path (J hook) of projectiles in soil.
f. It provides for proximity, contact, and delay fusing.
g. It models the vulnerability of personnel protected by structures and provides for user-specified vulnerability graphs for components protected by structures.

The main objective of this program development effort is to provide the designer of protective structures and the military strategist with specifications which relate hardness to the number of shots per attack and the number of attacks. By this means, it is anticipated that more economical (time or money or both) designs can be developed for various combinations of shots per attack, numbers of attacks, and weapon intensity. In like manner, a more real-world-oriented assessment of "enemy" structures will result from assessing hardness in terms of the probability of surviving various numbers of shots from a particular weapon system.



PREFACE

The research and development effort which produced the computer program described in this report began in 1976. The initial work was done by Falcon Research and Development Company under Contract No. DACA39-76-C-0009 to the U. S. Army Engineer Waterways Experiment Station (WES). The program development was partially completed when the contract with Falcon Research and Development Company was terminated in 1977.

Following the contract termination, the research and development effort was pursued at WES where the project was successfully concluded. This effort was sponsored by the Office, Chief of Engineers, U. S. Army, and was performed under Project No. 4A762719AT40, Task A1, Work Unit 003, and Task A4, Work Unit 001.

The principal investigator, Mr. Robert E. Walker, was assisted by Dr. J. L. Kirkland (until his retirement in 1977) and Ms. Barbara Robinson, all of the Structures Division (SD) which was reorganized in 1978 into the Structural Mechanics Division (SMD) of the Structures Laboratory (SL). Mr. William L. Boyt of the Automatic Data Processing (ADP) Center, WES, most ably assisted in correcting errors in the logic and in converting the program to operate on the WES G635 computer.

During the final phase of this effort, Mr. James B. Cheek, Jr., of the ADP Center (later of SD and then SMD) assisted in isolating faults, developing corrections, and enhancing the capability of the program. He also prepared this report and in so doing used portions of an unpublished draft of a contract report prepared by Messrs. R. K. Miller, T. J. Byrne, D. E. Hutchinson, and T. E. Dunning of Falcon Research and Development Company.

This work was accomplished under the technical leadership of Mr. Walker; the direct supervision of Mr. James T. Ballard, Chief, SD and SMD; and the general supervision of Mr. William J. Flathau, Assistant Chief, SL, and Mr. Bryant Mather, Chief, SL.

Commanders and Directors of WES during the work and the preparation of this report were COL J. L. Cannon, CE, and COL N. P. Conover, CE. Technical Director was Mr. F. R. Brown.

CONTENTS

CONVERSION FACTORS, INCH-POUND TO METRIC (SI) UNITS OF MEASUREMENT

Inch-pound units of measurement used in this report can be converted to metric (SI) units as follows:

Multiply	By	To Obtain
cubic inches	16.387064	cubic centimetres
feet	0.3048	metres
feet per second	0.3048	metres per second
foot-pounds (force)	1.355818	newton metres
grains (1/7000 lb mass)	0.06479891	grams
grains (1/7000 lb mass) per square inch	1.00438511	kilograms per square metre
inches	25.4	millimetres
inch-pounds (force)	0.1129848	newton metres
pounds (force) per square foot	47.88026	pascals
pounds (force) per square inch	6.894757	kilopascals
pounds (mass)	0.45359237	kilograms
pounds (mass) per cubic foot	16.01846	kilograms per cubic metre
pounds (nuclear equivalent of TNT)	2.092	megajoules
slugs (mass)	14.5939	kilograms
square feet	0.09290304	square metres
square inches	6.4516	square centimetres

DEVELOPMENT OF A PROGRAM FOR COMPUTER-BASED RESISTANCE ANALYSIS (COBRA)

PART I: INTRODUCTION

Purpose

1. This report describes a computer-based approach for evaluating the probability of survival, hence hardness, of protective structures and fortifications and the critical contents therein. Further, it documents the major mathematical and algorithmic procedures used by the program. And it provides instructions on how to prepare data for the program and illustrates the program's application to hardness evaluation of a protective structure.

2. This report also documents the limitations of the current approach and where appropriate recommends improvements needed to more closely model the actual behavior of weapons, fragments, the structure, and its contents.

Scope

3. The scope of this report is limited to describing:

   a. The approach used in modeling the structure and its contents.

   b. The method used to evaluate the survivability of the structure, its contents, and the composite thereof.

   c. The steps necessary to develop the data base.

   d. The computer-based procedures used to evaluate hardness in terms of probability of survival.

4. The report does not include judgments or recommendations based on the program's use thus far.

5. The program is limited in that each survival analysis is for the threat imposed by one type of conventional (nonnuclear) weapon. Additional analyses can be made for other weapons provided they are conventional designs.

6. The program's scope of application is broad in that it can:

   a. Account for impact, delay, and proximity fuses.

   b. Allow for variability in impact versus aim point.

   c. Identify structural components that are critical to the survival of the structure (i.e., columns, beams, etc.).

   d. Incorporate multiple layers of both critical and noncritical yet protective components of the structure (i.e., soil, sand bags, burster slab, etc.).

   e. Link critical components in such a way as to consider the enhancement to survivability due to redundancy in structure design.

   f. Evaluate damage, hence reduced survivability, due to blast from all weapons and due to fragments from proximity-fused weapons.

7. From the above, it can be seen that the scope of this effort is quite broad with respect to nonnuclear-generated blast and fragment threats. There are, as mentioned previously, areas needing further enhancement in order to more accurately model some important effects. These limitations are discussed in Appendix A.

Background

8. The history of hardness evaluation procedures is essentially contained in a statement attributed to a famous French leader: "That fortress can never be considered strong until it has once withstood full seige." However, such proof-tested structures are rare; also, such tests are very costly. Seeking less costly and more informative methods of hardness analysis, we have constructed physical models and subjected them to various intensities of attack in an effort to develop design criteria. Both actual combat and physical testing approaches have produced good results.

9. In spite of these successes, unresolved questions remain about why some structures stand when others fail. Equally important, if not more so, are the growing concerns for optimum use of resources in design, planning, and construction. Obviously, the best blending of time, money, personnel, and protection (hardness) is critical to

developing a good design. However, each planner, designer, and user may have a different view on what constitutes the most critical resources. For instance:

a. The battlefield commander facing imminent combat may see two kinds of time as critical. He must balance time to construct a specific hardness level against the delay time (resistance) that the position will offer in actual combat.

b. The planners working in a peacetime environment may see money as the most critical resource. Each dollar buys just so much protection. To them, stronger than necessary is wasted money. Even worse, too strong at one point means there will be a weakness somewhere else because there was not enough money left to build in the necessary hardness.

c. The small combat unit moving into a new position may see personal safety (hardness) as the most critical factor followed by the time required to produce that hardness with the few available personnel.

10. It might at first appear that an extensive physical testing program would produce enough data to answer all the questions on hardness. In one respect, this is correct. The difficulty lies in the interpretation of "extensive" testing. Extensive in this case means to subject every practical protective structural component and all practical combinations thereof to attack by every weapon against which they should be secure. Even this is not enough. Each weapon must be used repeatedly to account for the fact that no two weapons of the same type, aimed at the same point, or even delivered to the same point, will produce the same kind of damage. In short, there appear to be no simple safe versus unsafe answers about hardness. The specification appears to be largely probabilistic rather than deterministic.

11. The complexity and probabilisitic nature of the physical testing problem is well illustrated by a field testing experience in 1976. A small, rigid frame, fabric-covered structure was being evaluated for hardness when subjected to fragments from shells. On the first shot, one of the fragments perforated one of the supporting columns (a small pipe). The structure remained intact. On the very next shot, a fragment perforated the same column at almost the same place. Had it

been 1 in.* left or right, the column would have failed. Several points are noteworthy:

a. This actually happened.

b. The chances (probability) of it happening are very, very small.

c. The structure gives effective protection.

d. The structure is not 100 percent hard. (How hard is it?)

12. The above-outlined situation clearly shows the need for more useful data on structural hardness. What is needed is a reliable estimate of the likelihood of a structure being safe under a specific attack. Only by repeated testing can the influence of experimental errors be minimized through the application of statistical methods.

13. Interestingly, the more reliable data and the conclusions drawn therefrom may at first encounter appear less useful than the traditional safe versus unsafe statements. This new measure speaks in terms of "probability of survival," thereby indicating some doubt as to actual hardness. This new measure is a more accurate statement about the true situation. It carries with it both the influence of cumulative damage and the uncertainty inherent in the real-world process of attack and fortified defense.

14. The cost of physical modeling via repeated construction and attack to destruction is too high for all but the most simple or most critical structures. Therefore, we continue to look for reliable and less costly hardness evaluating methods. One promising new approach is that of computer-based modeling and evaluation by statistical methods. The remainder of this part presents an overview of this approach as it is implemented in this program.

Overview of Computer-Based Hardness Evaluation

15. Using this computer program to evaluate a structure's hardness requires that the analyst:

---

* A table of factors for converting inch-pound units of measurement to metric (SI) units is presented on page 3.

a. Prepare a geometrical model of the fortification's components consisting of the structure, its foundation, shielding (soil, sandbags, etc.), and objects to be protected (equipment and personnel). The components are defined by triangles. The definition, a rather tedious task, is made somewhat easier by the use of several surface and solid model-generating programs.

b. Indicate, via a code number, the physical character of each component (soil, concrete, personnel, wood, etc.). Note that personnel are components in this sense.

c. Identify, again by a code number, those components which are critical to the survival and mission accomplishment.

d. Define how the critical components interact with respect to survivability. That is, what combination of components must be "killed" in order to compromise the mission. This step enables the program to account for the fact that taking out a redundant column will not stop the mission, but that removing another critical (i.e., not redundant) structural component will result in total structural failure.

e. Define the vulnerability of equipment protected by the structure by giving several points from a probability of kill versus fragment velocity curve. Note that such curves for standard structural elements and personnel are built into the program.

f. Define the weapon in terms of its type (armor-piercing, high-explosive, etc.), fusing (contact, proximity, etc.), explosive charge, weight, fragmentation characteristics, and impact velocity.

g. Define the aim point and the azimuth and elevation angles of the entry path together with the width and length of the error ellipse.

h. Specify how many attacks will be made on the target.

i. Specify how many rounds will be fired in each attack.

16. With this basic information, the computer program subjects the structure and its contents to attack. The sequence of events is outlined below:

a. Calculate where the weapon will actually hit. This is accomplished using computer-generated pseudorandom numbers so that successive impact points will be normally distributed about the aim point. This correctly models the error inherent in weapon delivery systems. Since the size of the error ellipse is set by the user, it can be made small to simulate close-in shots or high-precision delivery systems or both (e.g., a guided

missile). The ellipse can be made large to simulate extreme range shooting or low-precision delivery or both (e.g., parachute-dropped).

b. Move the projectile along its entry path toward the impact point until it:

(1) Stops due to the resistance offered by the structure or the surrounding material or both.

(2) Passes through the structure.

(3) Misses the entire area modeled.

(4) Explodes due to satisfaction of the fusing requirement.

c. Calculate which components were involved in blast damage.

d. Evaluate the damage done by the blast and:

(1) Place blast-induced craters where appropriate.

(2) Evaluate the probability of survival of the structure as the result of this blast.

(3) Calculate the probability of surviving this attack by considering the blast-induced damage. Note that the cumulative damage assessment is in terms of survival probabilities of this and the preceding blasts.

e. Evaluate the damage done due to fragments and calculate the probability of surviving this attack due to fragment damage produced by all the fragment hits during this attack.

f. Calculate the composite probability of survival due to both blast and fragment damage.

g. Repeat this procedure until:

(1) All rounds for this attack have been fired, or

(2) The composite probability of survival is very near zero.

At this point, the probability of survival results have very little statistical significance. This is due to the fact that the results of single attacks on real or computer-modeled structures are unreliable indicators of hardness.

17. To overcome this difficulty, the program "rebuilds" the structure. It saves for future use the survival probabilities calculated for this attack. It then subjects the rebuilt (new) structure to another attack. This process is repeated until all of the

user-specified attacks have been made.

18. It may at first seem pointless to repeat the attacks. We might expect to get the same results each time, but such is not the case. Recall the statement made previously about the error ellipse, aim point, and the use of random numbers to select the impact point. Because of this procedure, the impact point will wander about the aim point during successive firings. Therefore, the damage from each round will most likely differ from that induced by any other round in the series of attacks. Note that this is not to say there will never be identical damage; only that this is quite unlikely (e.g., it is possible, but unlikely, to flip 20 "heads" in succession with a fair coin and coin-flipping device).

19. The hardness evaluation is completed by using the survival probabilities of each attack to compute the final hardness specification in terms of survival probability. Confidence in this number is justified because it comes from repeated tests in which the only variation is due to the random error inherent in the real-world process.

PART II: SIMULATION OF FORTIFICATION AND ATTACK

Coverage

20. This part describes the numerical, mathematical, and logical procedures used to model the structure and simulate its response to attack by nonnuclear devices. While it provides a conceptual description of the program, it does not focus on the fine points of logical, mathematical, and computer program analysis that might be needed by persons seeking to thoroughly master the program's logic. Those needs are addressed in the contractor's report,* from which some of the following material was extracted.

Definition of Position Coordinates

21. All positions are defined with respect to the origin of a right-hand Cartesian coordinate system in X, Y, and Z. This system is oriented such that the positive Z axis points up. An observer looking toward the origin from the positive end of the X axis will see the positive Y axis as horizontal and directed to the right.

Direction of Lines, Rays, etc.

22. In addition to position, it is necessary to specify direction, such as the path of the projectile, a normal to a surface, the direction of a ray, etc. In this program, direction is specified in terms of two angles, and these angles are defined with respect to the ray. The direction of any ray is established by giving the azimuth angle and the elevation angle.

---

* R. K. Miller et al., "BUNKER: A Comprehensive Simulation Program for Survivability of Equipment in Protective Bunkers Under Conventional Weapon Attack," Falcon Research and Development Company, Denver, Colo., May 1977, unpublished report prepared under Contract No. DACA39-76-C-0009 to the U. S. Army Engineer Waterways Experiment Station, Vicksburg, Miss.

23. The azimuth angle, as shown in Figure 1, is the angle between the positive Y axis and the projection of the ray onto the X, Y plane. Positive angles are those which increase in a clockwise direction when the X, Y plane is viewed from the positive Z axis (i.e., looking down toward the ground).

24. The elevation angle, as shown in Figure 2, is measured in the plane that contains the ray and is at right angles to the X, Y plane. The elevation angle is the angle between the X, Y plane and the ray as

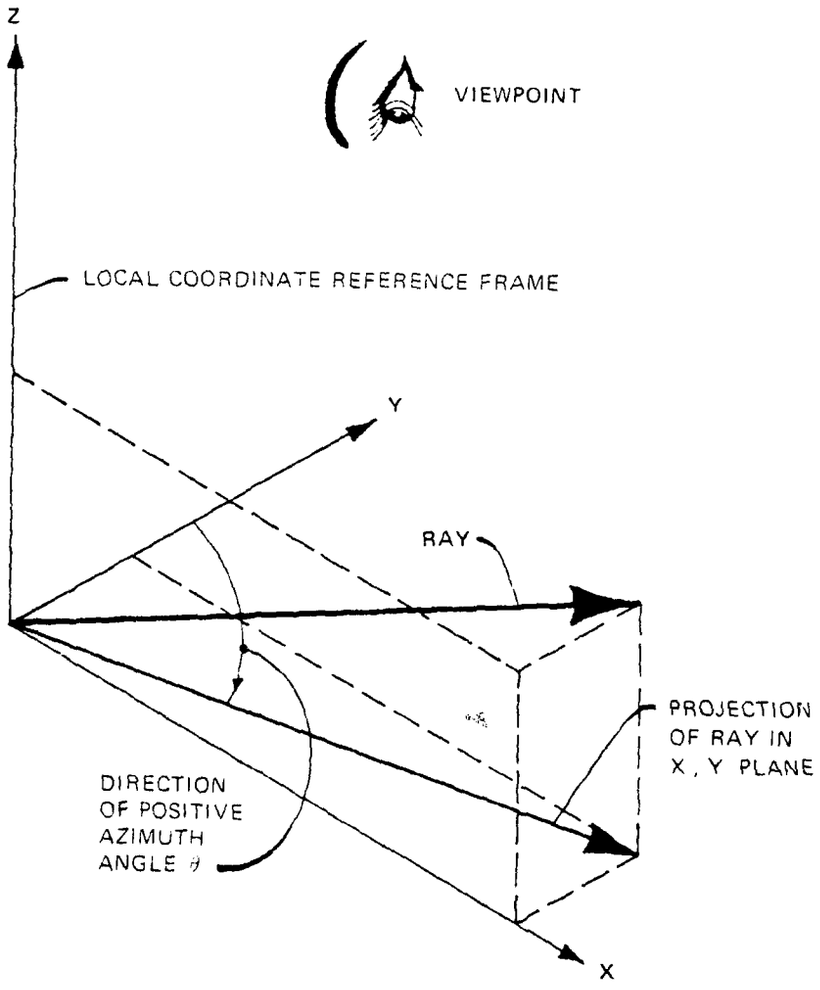


Figure 1. Azimuth angle

observed in the previously defined plane. The sign of the angle is positive when the ray lies above the X, Y plane. Note that, since the azimuth angle defines the horizontal direction, the elevation angle should not exceed $\pm$90 degrees because elevation angles greater than 90 degrees essentially reverse the horizontal direction given by the azimuth angle. For example, azimuth = 90 degrees, elevation = 135 degrees, is the same direction as azimuth = 270 degrees, elevation = 45 degrees.

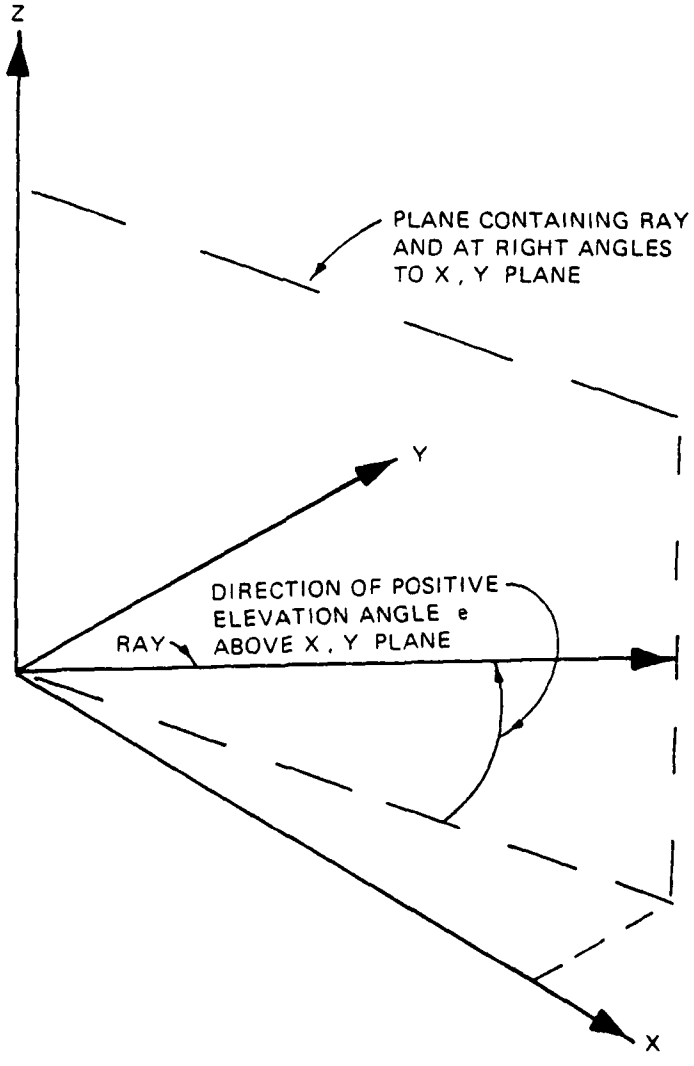


Figure 2. Elevation angle

Dimensions and Units

25. The dimensions of data supplied to the program are: time in seconds, length in inches, weight in pounds, angles in degrees, and probability [P(k/h)] in terms of incapacitating k after hitting h. With two exceptions, all other data are either dimensionless or unscaled combinations of these units. The exceptions, in deference to traditional weapons specifications, are that velocity is in feet per second and fragment weight is in grains (7000 grains = 1 pound).

Fortification Modeling

Components

26. The term "fortification" in the remainder of this report includes the structure, foundation, overburden, and surrounding soil plus the items of equipment and personnel being protected thereby. In the remainder of this report, the term "component" is used to define specific elements that comprise the fortification. For example, a radio, man, beam, plate, and sandbag are all components.

Describing components

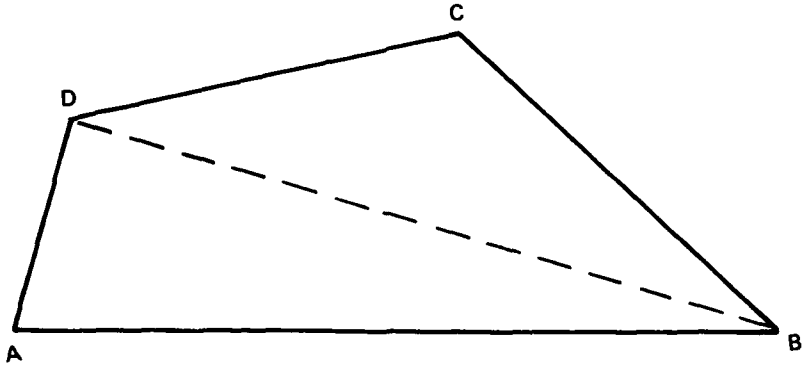
27. A component is described with a collection of triangles that cover its outside surface. A unique procedure for defining the triangles assures that the computer will correctly interpret the specification. Users do not normally prepare the triangular description. Instead, they take advantage of component-generating programs that prepare the required triangular description data for simple shapes (boxes, balls, cones, etc.). Complex components can than be assembled by combining such parts. However, insight into how the program operates can be gained by reviewing the component surface definition procedure. The following paragraphs present the definition requirements and illustrate their application in describing a simple surface.

28. The surfaces are defined by giving a list of X, Y, and Z coordinates of points. The points must be ordered in such a way that any three points define either a triangle or a line segment (one of the

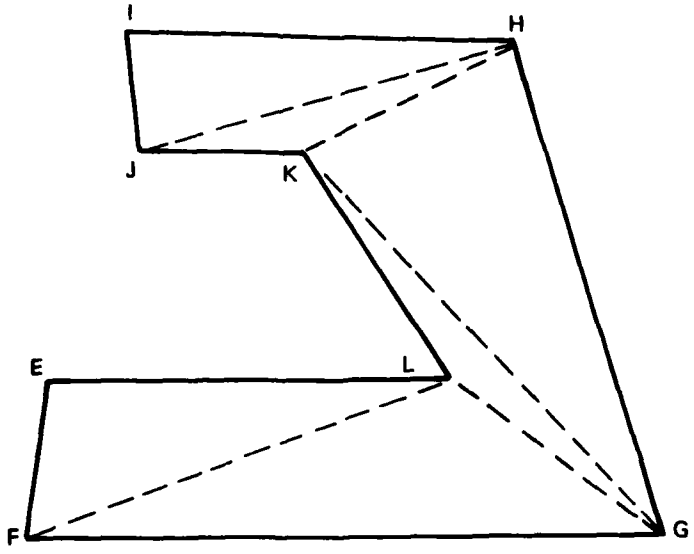
points being given twice). The triangles must not overlap each other.

29. This compact way to define surfaces is quite flexible in that complex shapes can be defined. Also, previously defined shapes are rather easy to alter; this feature is used to model the destruction due to blast effects. However, the actual process of defining surfaces according to the above rules can be quite complex for all except simple surfaces, as the following examples illustrate.

30. Figure 3 illustrates two surfaces to be defined. The letters



a. Simple surface defined by AABDCC



b. Complex surface defined by EEFLGKHJII

Figure 3. Two target surfaces defined

represent the points at the verticies. The first surface (Figure 3a) is subdivided into two triangles by a dashed line. The two triangles are defined according to the previously described procedure as AABDCC. Note that the first and last points are repeated (i.e., AA and CC), thereby indicating the first and last points used to define the surface or a portion thereof. This is a correct definition because the three points AAB define a line; the three points ABD and BDC define triangles; and points DCC define a line. The two triangles do not overlap. However, even this simple surface can be improperly defined by choosing the wrong starting point (e.g., BBDCAA). Here, the triangles BDC and DCA overlap. There is also a hole in the surface. Complex surfaces require even greater care in the choice of starting point and path. The definition of the surface shown in Figure 3b is EEFLGKHJII.

31. Such surface definitions seem obvious once they have been accomplished. However, the user need only try to define a few simple shapes without prior knowledge of the path to see the puzzlelike character of the procedure. The definition process is further complicated by the need to define the side, top, and rear faces of each solid component. For this reason, the previously mentioned target-generating programs have been developed. Their use removes from the user's consideration the restrictions of this essentially computer-oriented procedure.

Component types and properties

32. The components that define the fortifications are assigned by the user to one of the following categories:

a. Soil island. This is the rectangle of soil or material on or within which the fortification rests.

b. Critical structure. Critical structure components (beams, columns, plates, etc.) are those elements that contribute to the structure's failure if they are damaged.

c. Shielding. Shielding components (sandbags, burster slab, etc.) are those which protect the structure from damage but whose failure will not result in the failure of the fortification. Their loss or degradation simply reduces the hardness of the fortification.

d. Contents. These are components (personnel, equipment, etc.) which are being protected by the fortification and

are critical to carrying out the mission of the fortification. Note that noncritical structure components (i.e., partition walls, etc.) may also be inside (behind) the fortification along with the critical contents. They are treated as shielding (i.e., noncritical structure).

Weapon Delivery Simulation

Definition

33. The terms "weapon" and "projectile" appear to be uniquely defined only within a restricted segment of the weapons/fortifications community. Therefore, in the interest of precision and clarity, the term "projectile" is used throughout the remainder of this report to describe the device that is potentially the source of blast and/or fragments. It may also be capable of penetrating the components. In this broad sense, a bag of an explosive gas or an armor-piercing bullet is a projectile. The terms "weapon" and "weapon system" will be applied in this report to devices that can deliver projectiles.

Data required

34. Projectiles are described to the program in terms of their fusing (either contact, delay, or proximity) and their potential to produce blast- and/or fragment-induced damage. Blast damage potential is specified by giving the weight, in pounds of TNT, that would induce a blast equivalent to that of the projectile.

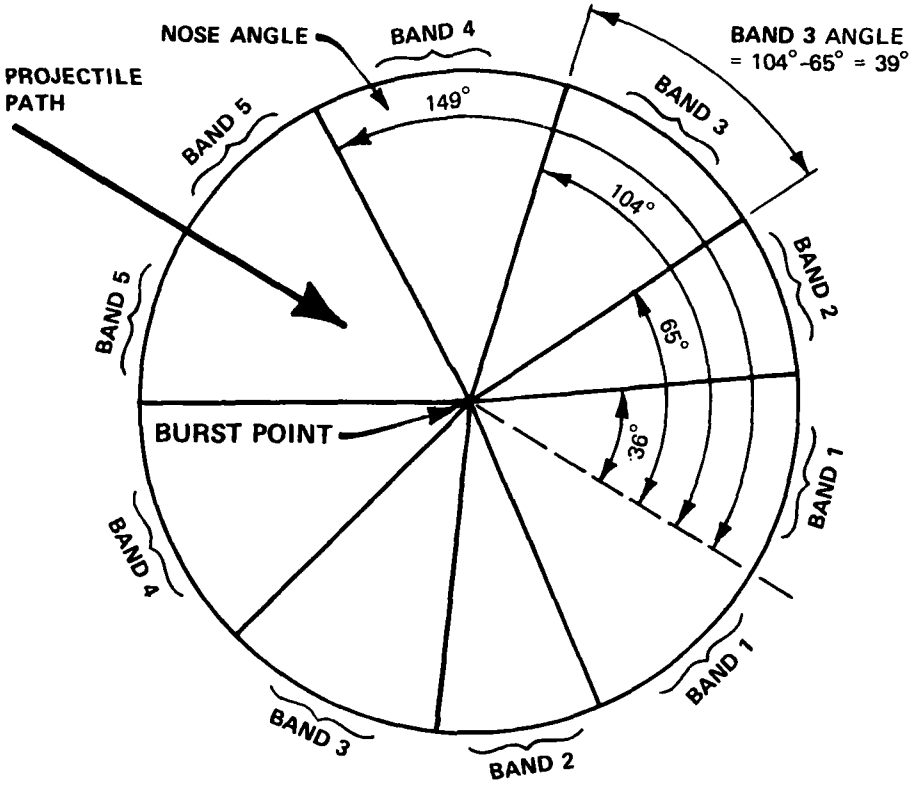
35. Fragment damage potential is defined in terms of the velocity, number, and weight of fragments in each angular band defined about the nose of the projectile. The definition conventions are illustrated in Figure 4. The user supplies the following fragment-related data for each nose band:

a. The nose angle, in degrees, for the upper limit of the band (the lower limit angle is taken as the upper limit of the previous band or zero for the first band).

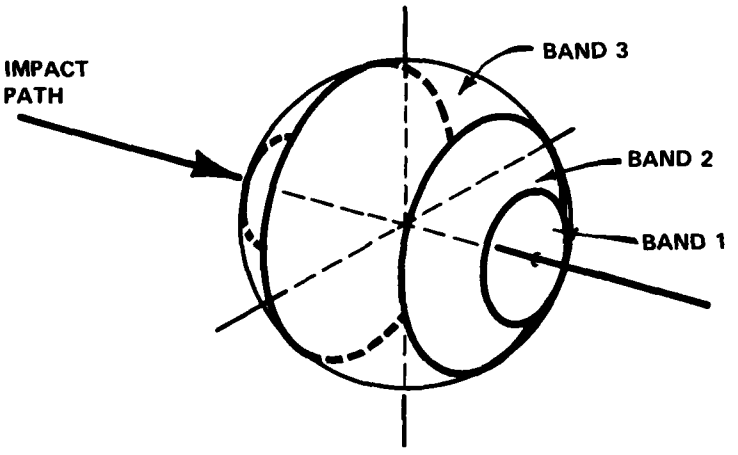
b. The mean velocity of all fragments.

c. The average weight values to be used in each of several different weight classes.

d. The number of fragments in each different weight class.



a. Side view



b. Isometric view

Figure 4. Fragment distribution bands

36. The program assumes that the several fragments in each weight class are uniformly distributed about the band. Consequently, a component which subtends 1/16 of a band will be hit by 1/16 of the total number of fragments of each weight. This in turn can result in a hit by fractional fragments (e.g., 1.6 fragments or 0.35 fragment), a situation which may seem wrong. However, in the overall sense, this is proper since the modeling process is essentially an averaging one involving statistics from many shots and many attacks.

Projectile delivery system

37. The projectile is directed in a straight line toward the impact point. The impact point lies in a horizontal plane at Z = 0. The point may or may not be coincident with a component. Thus, in this context, the term "impact point" does not necessarily imply contact with the fortification at that point. Instead, it means that the projectile will move along a line that passes through the impact point at the user-stated azimuth and elevation angles until it encounters some part of the fortification.

38. Note that for a given aim point all impact paths are parallel. This departs somewhat from the actual conditions of shooting from a fixed position wherein all impact paths originate at the same point (i.e., the weapon). This simplification is a practical limitation only when the distance between the assumed shooting position and the target is very small.

Simulation of weapon characteristics

39. The user supplies the impact velocity of the projectile, the length and width of the error ellipse, the aim point, and the azimuth and elevation angles of the arriving projectile's entry path (e.g., a projectile shot toward the origin from a point 100 ft out on the positive X axis and 50 ft up on the positive Z axis has an "arrival azimuth" of 90 degrees and an elevation angle of 26.6 degrees). The program assumes that the length of the error ellipse lies along the azimuth line and that the width is at right angles to the azimuth line. The ellipse is centered at the aim point. The program then uses the aim point coordinates, the width and length of the ellipse, and two

pseudorandom numbers to calculate the coordinates of the impact point within the error ellipse. The computer-generated pseudorandom numbers are converted from a uniform distribution (any value equally likely) to a normal distribution (average value most likely) to correctly model the fact that most impacts are near the aim point. Mathematically speaking, the impact points will be normally distributed along the length and width of the error ellipse.

40. As mentioned previously, the projectile is delivered to the impact point along a straight line (entry path) whose azimuth and elevation angles are stated by the user. Thus, the first decision the program must make is where the path begins. The path should begin far enough away from the aim point to assure that it is outside of the fortification. At the same time, it should not begin too far away from the fortification because the program will spend a great deal of time searching for intersections far beyond the fortification.

41. The program approaches these problems by finding X and Y coordinates on the impact path at which Z is slightly above the highest elevation on the fortification. This becomes the starting point of the impact path. The method assures that the starting point is outside the fortification. However, it imposes the requirement that the entry path elevation angle be greater than zero; thus, shooting at or below the horizon is not allowed. Further, shots very near horizontal will begin much too far from the fortification for computational efficiency. The difficulty here lies in the fact that the program moves along the impact path in short increments (user-supplied), testing at each increment for an intersection with one of the components. A lot of computer effort could be spent looking for intersections a long distance away from the fortification (this may not be a major concern in most field fortification simulations).

Correcting fragment distribution

42. The fragment distribution data supplied to the program are typically developed from static (zero projectile velocity) firings. These data specify bands whose angular limits are defined by the nose angle between the impact path and the dividing line between adjacent

bands (see Figure 4). The fragments within each band are assumed in this program to be projected at the single velocity specified for that band. Since the projectile velocity is rarely zero, the magnitude and direction of the original band velocity vector must be corrected to account for the projectile's velocity (e.g., a fragment leaving the static test burst point at 1000 ft/sec at a nose angle of 90 degrees will be traveling at 1414 ft/sec with a nose angle of 45 degrees when the projectile's velocity is 1000 ft/sec). The correction is accomplished by adding vectorially the projectile's velocity to the velocity vectors specified for the bands on each side of a given nose angle. The directions of the two resulting vectors are averaged to obtain the corrected nose angle (thereby avoiding a gap in the fragment distribution). The new velocity for the band is the average of the corrected velocities at the original upper and lower limits of the band.

Calculating projectile-fortification impact

43. The point at which the projectile actually contacts the fortification is determined by moving it in short increments along the entry path. During this movement, the velocity of the projectile remains as stated by the user (i.e., air drag is ignored). The movement is continued until the fuse is activated or the projectile has missed the entire fortification. Note that the first step in this process is finding the intersection of the impact path with the nearest component. Once an intersection is found, the fuse type determines subsequent actions. These actions are described in the following paragraphs.

Burst point for proximity-fused projectiles

44. The burst point for proximity-fused projectiles is located by backing along the entry path a user-specified burst distance away from the first intersection. Note that this is rather coarse modeling of proximity fusing wherein bursting is actually initiated by surfaces that enter the fusing sphere whose radius is the aforesaid burst distance. However, the programming and calculations needed to find the nearest

point on the nearest surface may not be justified in many practical problems (see Appendix A).

45. Another limitation that has no direct relation to but nevertheless influences proximity fusing is that of minimum projectile velocity. The logic for delay-fused projectiles is such that burst is initiated when the velocity is less than 10 ft/sec. This rule also applies to proximity fuses. Therefore, the user must specify a projectile velocity of more than 10 ft/sec; otherwise, the burst will occur at the starting point of the entry path.

Burst point for
contact-fused projectiles

46. Contact fuses are simulated by setting the fuse delay time to zero. The burst of contact-fused projectiles is initiated at the first component intersection. Note that the minimum velocity limitation of 10 ft/sec previously described for proximity fuses also applies to contact-fused projectiles.

Burst point for
delay-fused projectiles

47. Delay-fused projectiles are assumed to operate only in soil or in air after encountering soil (e.g., a crater). Contact with any other material causes the remaining delay time to be ignored and the burst to be initiated at the contact point. The assumption is that the projectile will be slowed down by the structural component and thus remain therein long enough to activate the fuse. This assumption is questionable for thin structural elements.

48. When first encountering soil, the timer for the elapsed fuse delay time is started. As the projectile moves incrementally through the first and subsequent layers of soil, the velocity is reduced, the time required to penetrate each layer is accumulated, and the path through the soil is curved to simulate the hooking behavior of projectiles in soil. Projectiles that exit soil into air retain their exit velocity and move in a straight line. The time required to transverse the opening is accumulated as the elapsed fuse time. Projectiles may

reenter soil from air, in which case the path and velocity changes induced by soil will apply.

49. Projectile burst is initiated when the elapsed fuse time is greater than the fuse delay. Projectiles also burst when the velocity is less than 10 ft/sec. The mathematical formulas used in the path and velocity calculations are presented below.

50. The equations used were derived by assuming that the force acting on the projectile was a drag force F_D, of the Poncelet form, as defined in "Terminal Ballistics,"*

$$F_D = m \frac{dV}{dt} = -A(a + bV^2) \tag{1}$$

where

m = mass of the projectile

V = projectile velocity

t = time

A = projectile cross-sectional area

a and b = empirical constants

51. This form was chosen because it appeared to be a valid model of observed soil penetration mechanics. Also, the empirical constants a and b can be chosen so that the derived relation for total penetration path length (i.e., the distance that the projectile has penetrated, measured along the projectile trajectory) corresponds closely to the values given in a National Defense Research Committee (NDRC) report** (Chapter 19, Figure 2A2, page 394).

52. Using S to denote penetration path length, and with everything else as previously defined, the needed derivations of Equation 1 are

$$S_2 - S_1 = \frac{m}{2Ab} \ln \frac{a + bV_1^2}{a + bV_2^2} \tag{2}$$

---

* M. E. Backman, "Terminal Ballistics," NWC TP 5780, February 1976, Naval Weapons Center, China Lake, Calif.

** National Defense Research Committee, Effects of Impact and Explosion, Vol I, Washington, D. C., 1946.

Rearranged in terms of V_2^2, Equation 2 becomes

$$V_2^2 = \frac{a + bV_1^2 - a\, e^{\left[\frac{2Ab}{m}(S_2 - S_1)\right]}}{b\, e^{\left[\frac{2Ab}{m}(S_2 - S_1)\right]}} \qquad (3)$$

53. The incremental change in the soil penetration calculation is in the penetration path length S rather than time. This appears to be the more logical way to proceed since concern is with the position of, and the path taken by the projectile. Therefore, with the changes in the path length $(S_2 - S_1)$ specified, Equation 3 is used to determine the new velocity V_2 at the end of a path increment.

54. Total penetration path length S_T is obtained from Equation 2 by setting $S_1 = 0$, $V_2 = 0$, and $V_1 = V_i$, where V_i is the projectile impact velocity. The resulting equation is

$$S_T = \frac{m}{2Ab} \ln\left(1 + \frac{b}{a} V_i^2\right) \qquad (4)$$

55. With the incremental velocity change known, the incremental change in time t is obtained by direct integration of Equation 1. The result is

$$t_2 - t_1 = \frac{m}{A\sqrt{ab}} \arctan V_1\sqrt{\frac{b}{a}} - \arctan V_2\sqrt{\frac{b}{a}} \qquad (5)$$

56. The equation used to determine the orientation of the projectile, and to help determine its path in three dimensions, is similar to the one explained in "Terminal Ballistics." The equation assumes that the deflecting force is a component of the total drag force and relates it to the centripetal force felt by a body following a curved trajectory. The deflection angles are defined as shown in Figure 5.

57. If r is the radius of curvature of the projectile trajectory, then

$$\frac{d\Omega}{dS} = \frac{1}{2r}$$

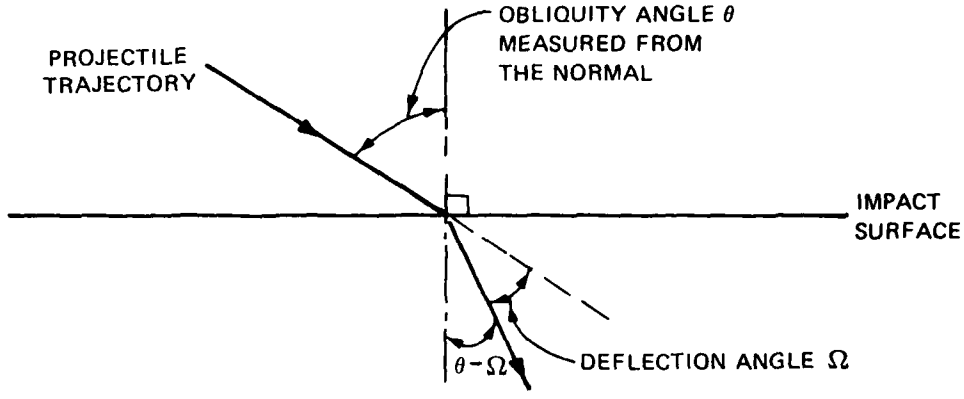


Figure 5. Angles used in projectile deflection calculations

The deflection force F_d is

$$F_d = -A(a + bV^2) \sin (\theta - \Omega)$$

The centripetal force is

$$F_d = \frac{mV^2}{r}$$

Therefore,

$$\frac{d\Omega}{dS} = \frac{F_d}{2mV^2}$$

and

$$\frac{d\Omega}{dS} = \frac{-A(a + bV^2) \sin (\theta - \Omega)}{2mV^2}$$

Rearranging this equation gives the desired result

$$\frac{d\Omega}{\sin(\theta - \Omega)} = \frac{-A}{2m} \left(\frac{a}{V^2} + b\right) dS$$

The instantaneous deflection is

$$\Delta\Omega = \theta - 2 \arctan \left(\frac{\tan \frac{\theta}{2}}{e^{1/4} \left\{ \ln \left[\frac{a + bV_1^2 - a\, e^{\left(\frac{2Ab}{m} \Delta S \right)}}{bV_1^2} \right] - \frac{Ab}{2m} \Delta S \right\}} \right) \quad (6)$$

Changing from the angular specification (see Figure 6) to coordinates is accomplished by

$$X_2 = X_1 - \Delta S \cos \alpha \sin \nu$$

$$Y_2 = Y_1 - \Delta S \sin \alpha \sin \nu$$

$$Z_2 = Z_1 - \Delta S \cos \nu$$

where X_1, Y_1, and Z_1 define the previous X, Y, and Z coordinates, X_2, Y_2, and Z_2 are the new endpoint coordinates of the segment ΔS long, and α is the azimuth angle.

58. The values of the empirical constants a and b vary

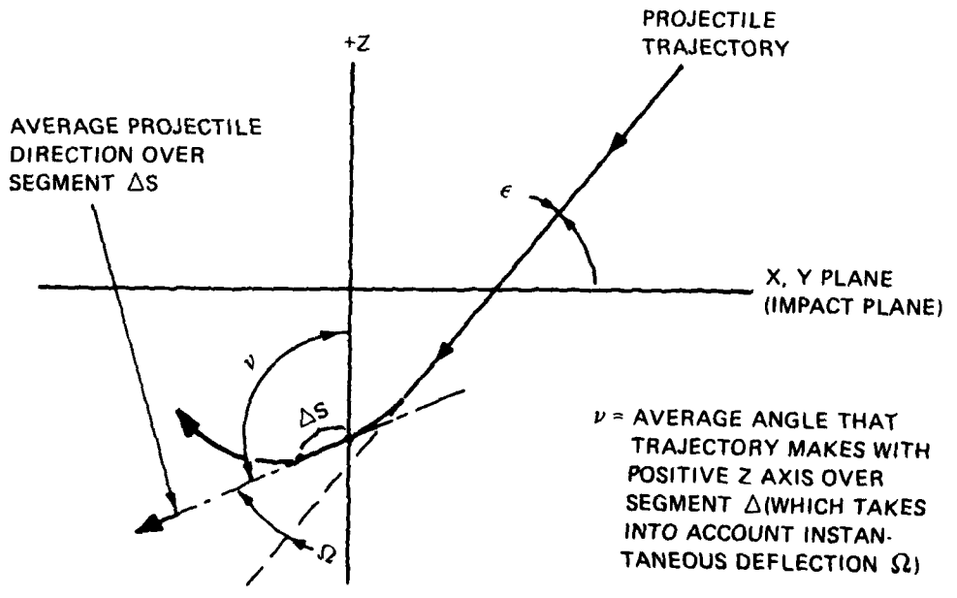


Figure 6. Angles used in projectile trajectory calculations

according to the projectile's nose shape, caliber density (defined as the weight of the projectile in pounds divided by the cube of the diameter expressed in inches; i.e., wt, lb/(diam, in.)^3), and the type of soil that the projectile is passing through. Both a and b can be determined from

$$a = k_1 \rho_c^{2/3}$$

$$b = k_2 \rho_c^{2/3}$$

where k_1 and k_2 are constants that take into account the projectile's nose shape and the type of soil that the projectile is in and ρ_c is the caliber density of the projectile. Using a variety of caliber densities, k_1 and k_2 were determined with a least-mean-squares routine that determined the best fit that Equation 4 made with the curves in Figure 2A2, page 394, of the NDRC report.

Attack Simulation

Finding the threatened components

59. In order to model the interaction between the burst and each component, the program must first decide which components are threatened. This is accomplished by generating rays in many directions from the burst point. For each ray, the program calculates its intersection with every component that lies on that ray. It arranges the components in increasing order with respect to distance from the burst point. By this means, the program can subject the nearest component to the simulated blast and fragment damage, and, if the component is defeated (removed or perforated), the program will proceed to attack the next component.

60. The concern here is not for the attack/defeat process; this will be discussed later. Instead, it is for how the rays are generated. Obviously, a limitless number of rays would be ideal in that every component would be "found" (i.e., intersected by a ray). Unfortunately, the computer's memory places an upper bound on the maximum number of

rays (currently about 3000). The following paragraphs describe how the program and the user interact to produce an acceptable distribution of rays. The objective of this interaction is to have at least one ray pass through each critical component. Better yet, several rays should pass through each critical component. To accomplish this, several user specifications are needed to help generate the ray bundle.

Large grid/small grid rays

61. The program provides two ray densities, large grid and small grid, for use in modeling component-blast-fragment interaction. For each critical component, the user must specify whether it is to be modeled with a large or a small grid. If a large grid models a component, then fewer rays will pass through that component than would were it modeled with a small grid. Consequently, the small grid specification is given for small-sized critical components in an effort to force at least one ray to pass through them. The small grid specification would also be given for a large component whose complex shape needs a fine (small) grid to define its limits. The large grid specification is used for large critical components whose shape is simple (i.e., square, rectangular, etc.).

62. It is important to note that "large" and "small" in this context refer to the sizes of the triangles used to define the component. In this respect, a "large" cube defined in terms of many small cubes is considered "small" because it is described with small triangles. A "large" cube defined in terms of its basic triangles is considered to be "large" because it is actually defined in terms of large triangles.

63. User judgment based on knowledge of the fortification and experience with the program will help in determining which grid specification to make. If in doubt, note that using the small grid specifications for all components will produce the finest ray definition along with a significant increase in the computer time required to process the problem.

Problem-generated
large and small grids

64. Each of the rays from the burst point passes through the

centroid of one of the elements of the small grid or the large grid. That is, a ray is generated for each small and large grid element. These grids are generated by the program using two values supplied by the user. One value specifies the number of segments in the small grid. The other value specifies the number of segments in the large grid. Since small and large grids are identical with respect to generation, the following description applies to both. The only difference will be in the angular (azimuth and elevation) dimensions of each grid element.

65. The program generates the ray grid by dividing the surface of a unit sphere (radius = 1) into $2N\left[\frac{(N-2)}{4}+1\right]+2$ patches, where N is the value given by the user for the size of the grid. This is accomplished by dividing the azimuth circle (360 degrees) into N parts as illustrated in Figure 7. Note that each azimuth division subtends an equal azimuth angle.

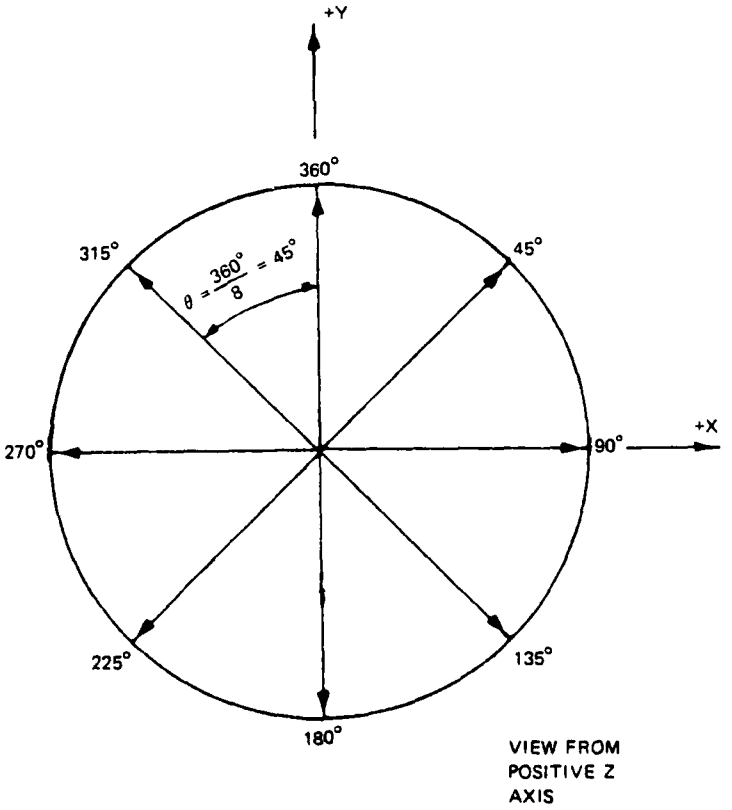


Figure 7. Ray grid with azimuth circle for N = 8

66. Such is not the case for each elevation increment. Instead, the elevation is subdivided into $2\{[\ (N - 2)/4] + 1\}$ equal parts of sin θ where θ varies from -90 to +90 degrees; thus, sin θ varies from -1 to +1 (Figure 8).

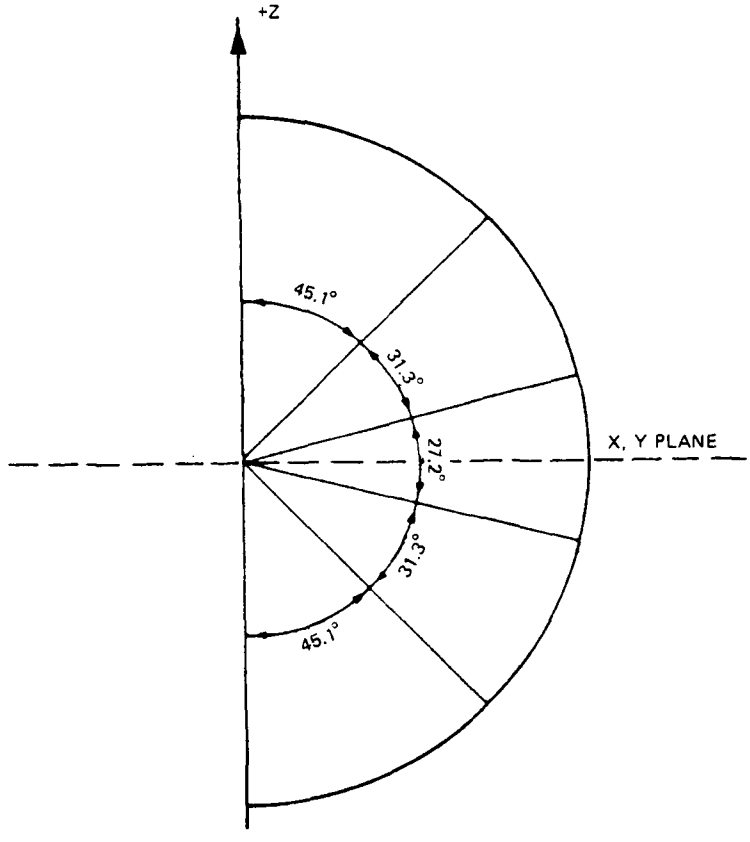


Figure 8. Elevation subdivision for use with eight azimuth subdivisions

Ray selection

67. The purpose of the small and large grids previously described is to generate candidate rays that start at the burst point and pass through the centroid of each grid element. The term "candidate" indicates that not all these rays will be used. Only those that meet the following conditions will be selected:

a. The ray must pass through a critical component having a

matching grid size specification (i.e., small grid components are only tested against small grid rays; likewise for large grid components/rays).

b. The maximum angular limits (azimuth and elevation) of the smallest rectangle that bounds the triangle (part of a component) being tested must be greater than the corresponding angular limits of the grid element associated with the ray. That is, the triangle must be greater in both azimuth and elevation dimensions than is the projection of the grid element onto the triangle. In short, the triangle must be "larger" (in the above sense) than the ray's grid element.

68. This selection process eliminates those rays that miss the target entirely, those that only pass through noncritical components, and those that only hit "small" parts of components.

Additional rays

69. Since small components as well as components made up of small parts may be missed by the selection procedure, additional rays may be needed. These are developed by generating new rays that pass through the centroid of each component. The new rays are added to the previously selected group. The grid and centroid developed rays are illustrated in Figure 9. The candidate rays which were not selected are not shown in this figure. Note that the large grid is shown as being closer to the burst point than is the small grid. This is for illustrative purposes only. The grids are properly visualized as being projected onto each component.

Deleting unnecessary rays

70. The selection and addition process can develop rays that are nearly identical, differing only slightly in azimuth and elevation. In order to reduce the unnecessary calculations associated with such rays, the program transfers the intersection on the large grid ray to the small grid ray of the near identical pair. In Figure 9, for example, rays U and T pass through the same small grid. Ray U will be deleted and the surfaces it intersects will be transferred to ray T.

Highlighting critical components

71. The components protected by the fortification may be widely dispersed from each other. On the other hand, there may be several

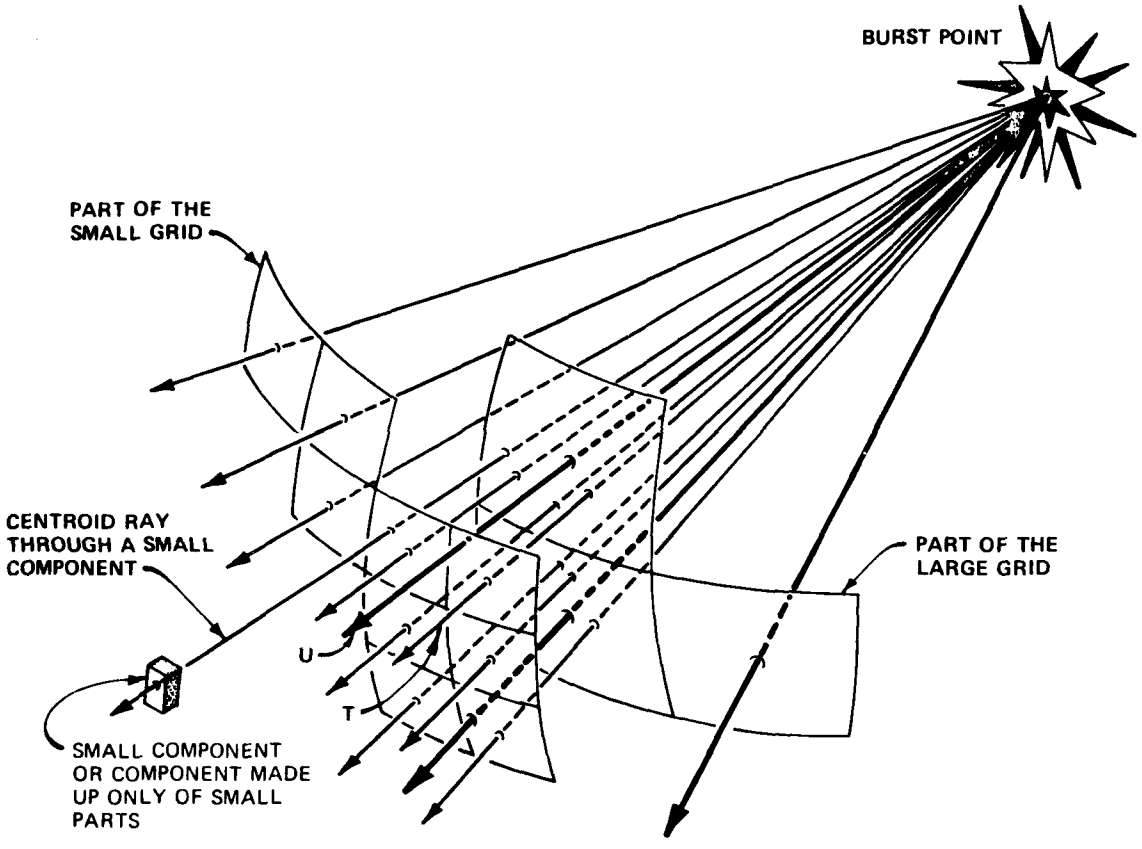


Figure 9. Illustration of ray selection and addition

or only one critical component. Since the burst may occur anywhere around or within the fortification, a component may be close to one burst and far away from the next. Thus, there must be some way to be sure that each one of the critical components is exposed to an appropriate fraction of the blast and fragment damage. This is accomplished by having the user highlight the important groups of components and/or fractional parts thereof by enclosing each group with a spherical shell called a "glitter sphere" (Figure 10).

72. Glitter spheres are defined in terms of the sphere's radius (glitter radius) and the sphere's center (glitter point). To identify and highlight each critical group of components, the user simply specifies a glitter point near the center of the components and gives a glitter radius value large enough to enclose a group of critical

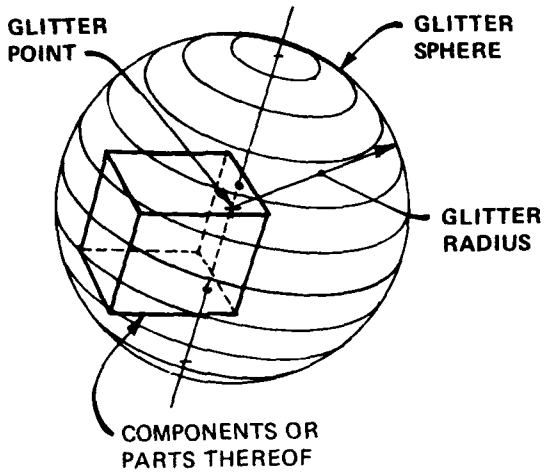


Figure 10. Highlighting important components with a glitter sphere

components within the resulting glitter sphere.

73. The glitter sphere's dimensions are normally only approximate. Thus, the glitter sphere may be slightly larger or smaller than the group of critical components and only approximately centered about them. Only in extremely complex targets involving several closely spaced glitter spheres should the user try to define the smallest glitter sphere that will contain each group. The following paragraphs describe how the program uses glitter data in modeling the threat to the components.

The role of glitter spheres

74. The glitter spheres are used to decrease the angular dimensions of the large and small grid elements (i.e., increase the number of azimuth and elevation divisions). Recall that these dimensions are initially set by the user via the specification about the number of small N and large M grid divisions.

75. The reason it is sometimes desirable to reduce the angular dimension (hence increase N and M) is illustrated in the following

example which considers only the azimuth angle and width (horizontal dimension). Assume the target is a 10- by 10- by 10-in. cube having one face at right angles to an azimuth line between two grid elements. In this example the number of small grid divisions is 100. Thus, the angular azimuth dimension of each grid element is 360 degrees/100 = 3.6 degrees. As illustrated in Figure 11, there will be 12 small grid elements across the face of the cube when the burst point is 12 in. from the target face. However, if the next burst point is 120 in. from the face, no grid elements will lie within the target face (Figure 12).

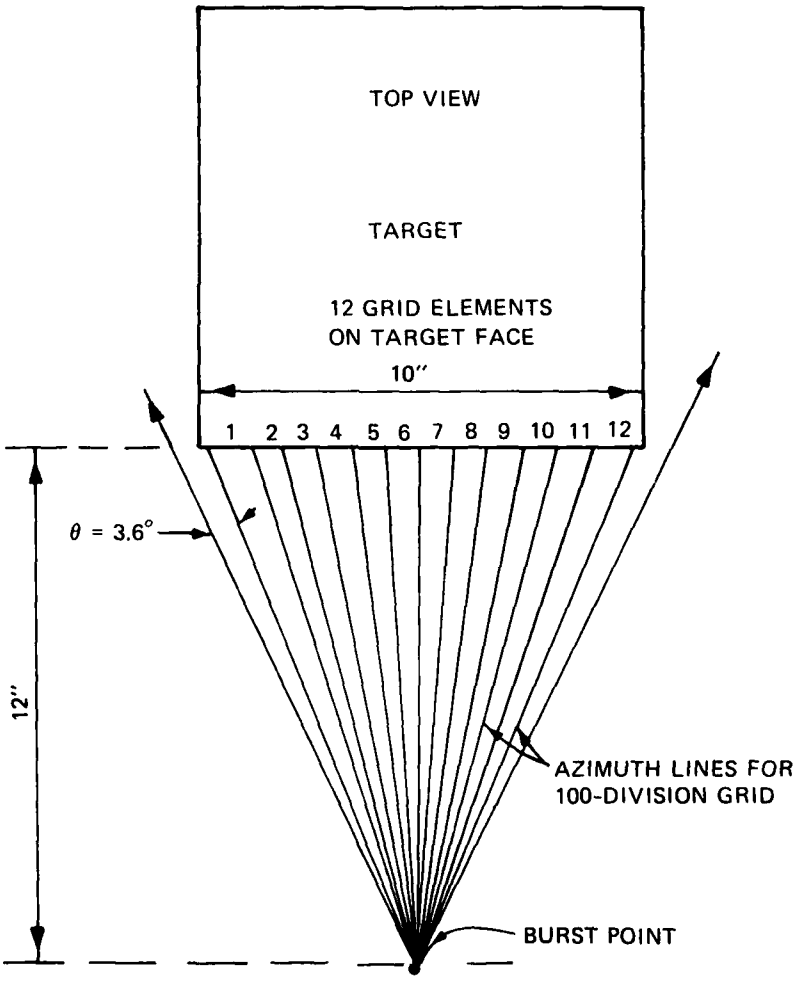


Figure 11. Grid intersections for a burst close to the target

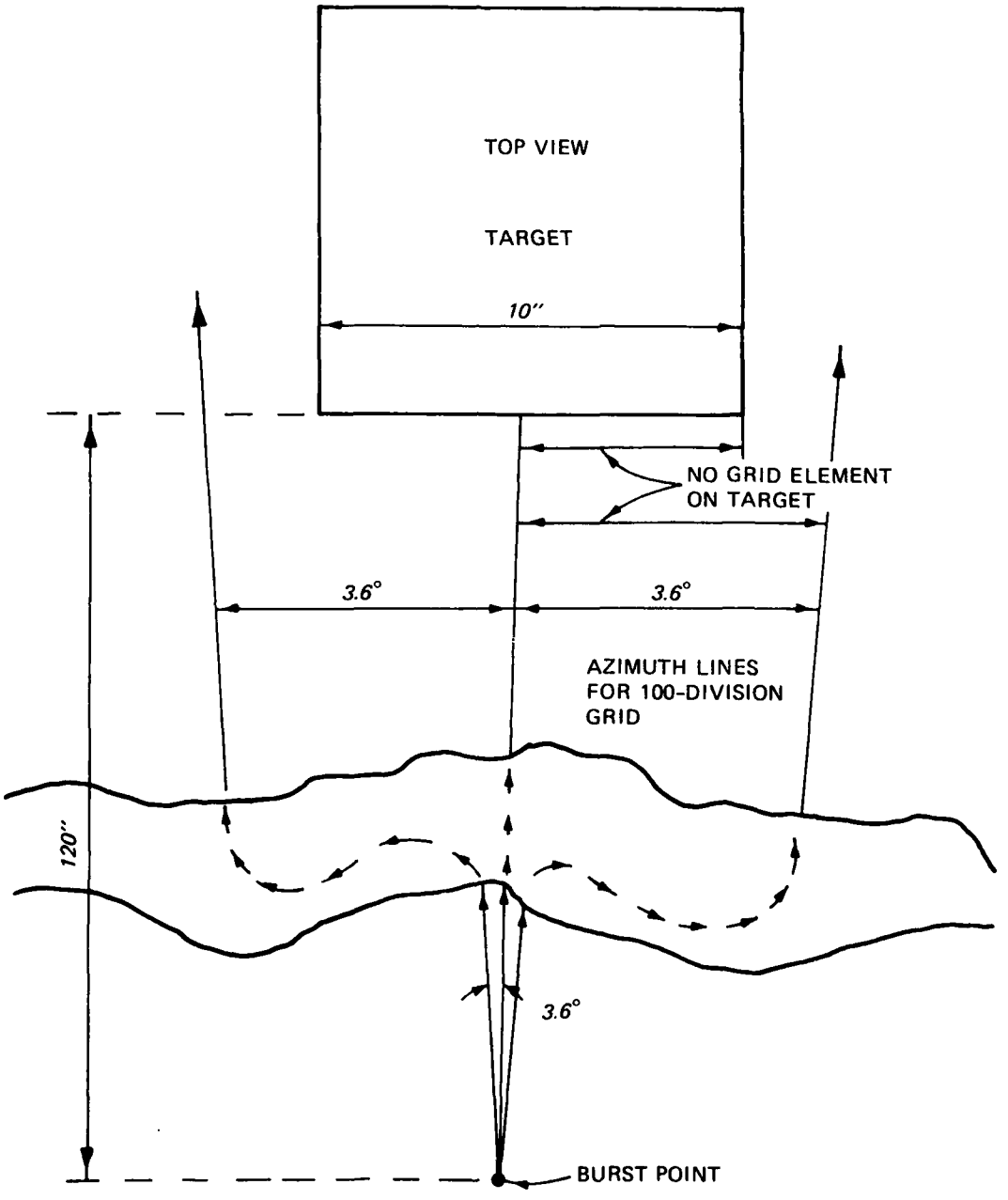
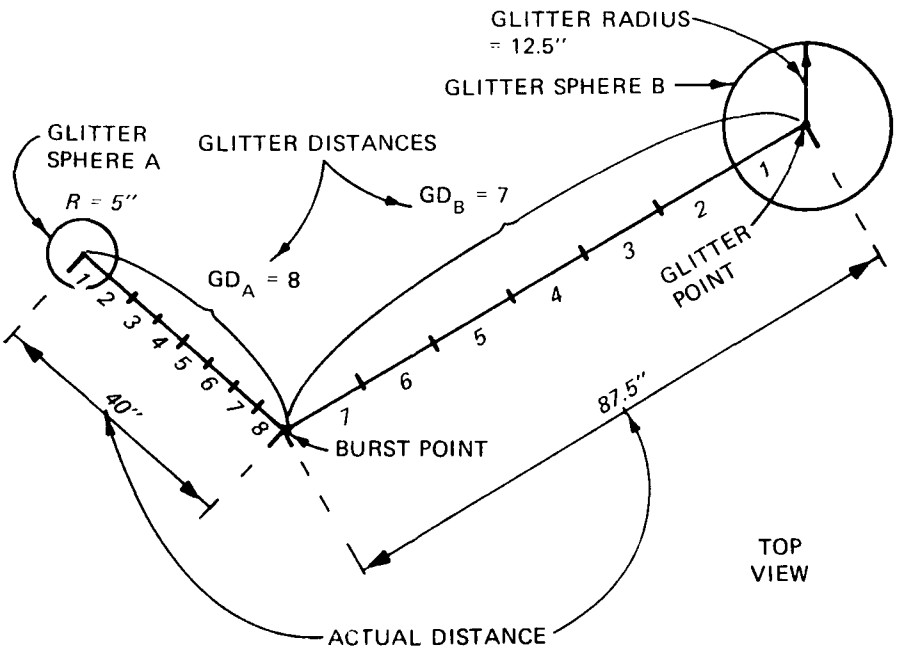


Figure 12. Grid intersections for a burst remote from the target

76. The glitter sphere is used to overcome the above-illustrated problem of having too few grid elements (hence rays) on the target. The program recognizes this problem by calculating the glitter distance to each glitter point. As illustrated in Figure 13, the glitter distance equals the actual distance from the burst point to the glitter point divided by the glitter sphere radius. The program then locates the largest glitter distance. If it is less than one, then the burst point is inside the associated glitter sphere, in which case the program applies (without change) the user-specified values for small and large grid divisions. Note that the user may defeat the glitter point influence by giving very large values to all glitter radii so that each burst point will be inside of every glitter sphere.



NOTE: A IS CLOSER TO BURST POINT THAN B, BUT A IS THE CRITICAL GLITTER SPHERE BECAUSE IT HAS GREATER GLITTER DISTANCE THAN B.

Figure 13. Computing glitter distance

77. In those cases where the largest glitter distance GD_{max} is greater than one, the program raises GD_{max} to the 0.6 power and multiplies the two user-specified values for small and large grid divisions by that result. Those two new values for small and large divisions will be used unless they exceed the maximum allowable value (currently 300). Values exceeding this maximum will be reduced to that maximum allowable value.

Vulnerability and Damage Simulation

Blast-induced damage

78. Blast damage is simulated with blast propagation models which calculate the effect on various structural elements. Keeping in mind possible previous damage, the program produces a current probability for failure of each element. It considers the interrelation of all critical structural components and combines their survival probabilities to produce a probability of survival for each shot. Note that blast-induced damage is not modeled for critical contents protected by the fortification. They are assumed to be otherwise protected or immune to blast-induced damage.

79. Several simplifying assumptions were made to calculate blast propagation and reduce the amount of computer storage needed to determine which elements in the structure have failed. The models for blast propagation are readily accessible and can be changed easily. The mechanical models would be difficult to modify substantially.

Applied force calculation

80. The force applied on a face or object of solid angle ω is calculated using the pressure-distance empirical relationship from Chapter 3 of the NDRC report

$$p = \frac{k}{\lambda^3}$$

where k is a constant, and

$$\lambda = \frac{r}{W^{1/3}}$$

$$p = \frac{kW}{r^3}$$

Since the area of solid angle ω is

$$A = \omega r^2$$

the force F on the object is

$$F = kW_{\omega}\int\frac{dA}{r^3}$$

$$\approx \frac{kW\omega r^2}{r^3}$$

$$= \frac{kW\omega}{r}$$

81. When the blast propagates from soil to soil with changed k, the true situation is quite complex. To approximate the true situation and yet provide a model which can use the rays from the point burst, the program ignores reflections at soil interfaces. When an interface between soils of types A and B is encountered, the distance to the back side of A is adjusted to be the right distance through type B to give the same force. That is,

$$r'_A = \frac{k_B}{k_A} r_A$$

$$r_B = r'_A + t_B$$

where t_B is the thickness of B, r_A and r_B are distances to the back surface of A and B, and r'_A is adjusted r_A. Thus,

$$F = \frac{k_B W\omega}{r_B} = \frac{W\omega}{\frac{1}{k_B}\left[\left(\frac{k_B}{k_A}\right) r_A + t_B\right]}$$

$$= \frac{W\omega}{\frac{r_A}{k_A} + \frac{t_B}{k_B}}$$

and, if $r' = \Sigma\ t_i/k_i$ is accumulated, where t_i is the thickness of the i^{th} soil and k_i the soil constant, then

$$F = \frac{W\omega}{r'}$$

Note that the earth shock computations ignore air gaps in the soil of less than 1 in. but assume no propagation across a larger gap.

82. To change the soil propagation model, we may either change the present processor or add a new material type such as

$$\rho = k\ e^{(a\lambda)}$$

where λ is $r/W^{1/3}$ and α and k are constants.

83. The air shock model is similar to the earth shock model; however, only one kind of air is included and shock is only transmitted to structural elements. The pulse length is currently set at 0.001 second. The model used for airblast is

$$p = \frac{7.12 \times 10^6}{\lambda^3} - \frac{1.51 \times 10^4}{\lambda^2} + \frac{474}{\lambda}\ (\lambda \text{ in inches/lb}^{1/3})$$

84. The force impinging on a structure which is transmitted through soil and/or air is added to the force vectors. Note that structural elements directly in contact with the first one on each ray and also on that ray are assumed to receive the same force as the first element. As mentioned previously intervening air gaps of less than 1 in. are completely ignored. Intervening soil of less than 5 in. is also ignored.

85. It should be noted that the total shock model was designed with underground bursts in mind. As a consequence, transmission of blast from air into soil was not considered. With air bursts, however, significant amounts of earth shock are produced if the burst is within about 1 scale foot of the surface of the soil. Therefore, the earth shock values produced by this program will be less than real world shocks.

86. After all of the rays have been processed for forces applied to each structural element, these accumulated forces are then used to determine if the protective structure still stands. Each type of material is treated differently, owing to different failure modes and cumulative damage effects.

87. Materials which deform elastically until failure are treated as harmonic oscillators responding to a forcing function. In the case of blast, the forcing function approximates a triangular pulse. From the pulse duration and peak pressure, the response can be calculated and from this, the maximum stress. The blast calculation finds the ratio of the maximum stress due to the shock to that which would be caused by a static force of the same magnitude. After the ratio is applied to the static value, the program performs a comparison with the maximum possible loading on the beams and produces a probability of kill which basically states that the beam has, has not, or possibly has failed. This probability is coupled with previous values to obtain a current estimate. A weakness in this algorithm is the way the total force on a beam is produced and then used to find the bending moment applied to the beam. If the force is not uniformly distributed, the moment will be incorrect. For example, Figure 14 illustrates various applied forces, the true bending moments, and the calculated moments. It can be seen that if the force applied to the beam is not too strongly concentrated in one spot the calculated bending moment will be acceptable.

88. Reinforced concrete is treated with the use of the graph in Figure 6A5, page 434, of the NDRC report. The graph as given, however, is somewhat restricted in scope. It applies only to reinforced-concrete walls buried in soil with $k = 5000$, subjected to earth shock from charges buried opposite the center of the wall. In recognition of these limitations the following adjustments are made.

89. Using the total force on the wall and the wall area, the average pressure is calculated. Since in the analysis that produced the graph it was assumed that the peak pressure on the wall could be decremented by 25 percent to give average pressure, the average actually

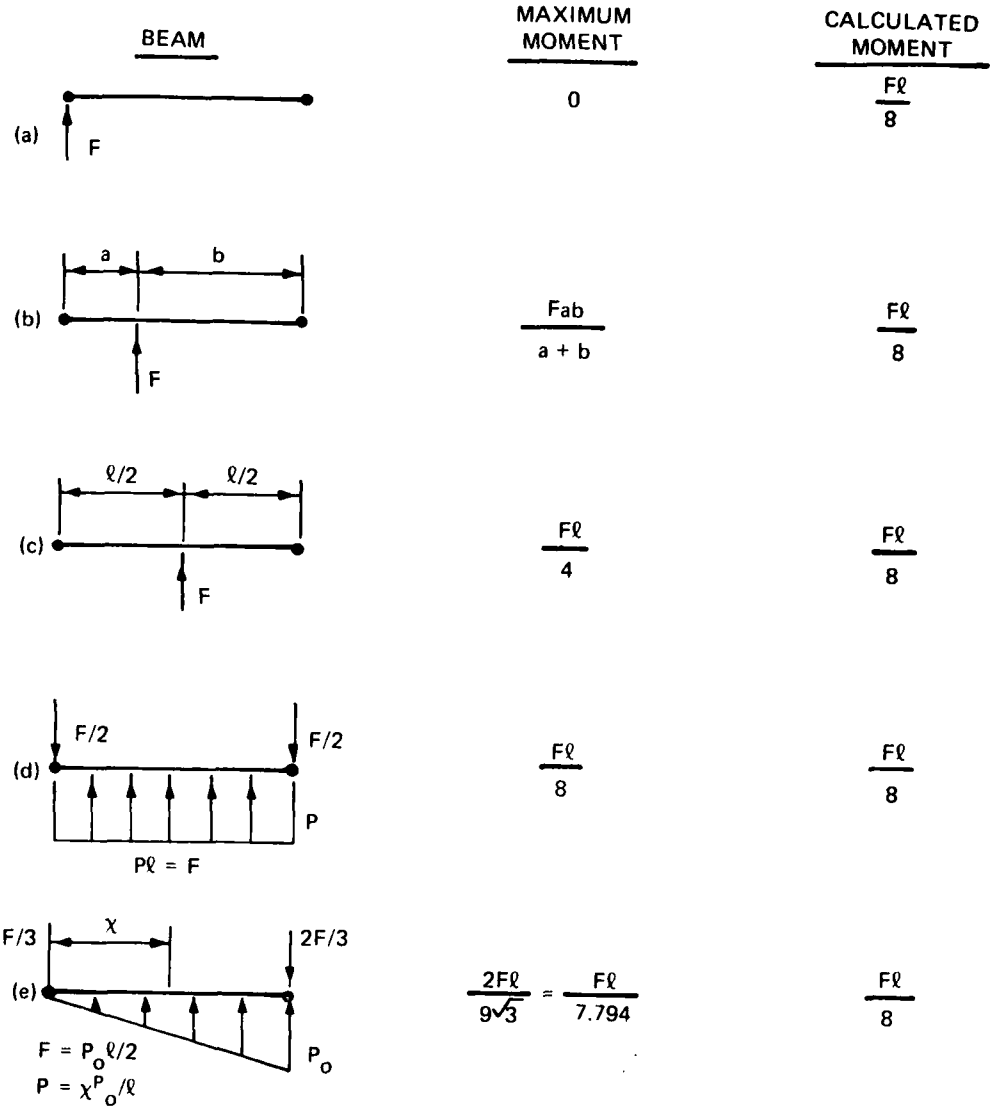


Figure 14. Example of applied force, true bending, and calculated moments

obtained is incremented by 30 percent giving, hopefully, peak pressure in the sense of the analysis. This peak pressure is then converted to a distance through average soil using the model

$$p = \frac{k}{\lambda^3}$$

$$\lambda = \left(\frac{k}{P_o}\right)^{1/3}$$

λ is then used with empirical fits to the above-mentioned graph to give a degree of damage and an associated probability of kill. The probability of kill indicates not so much a chance of failure, as it does a degradation of function. The successive increase of the kill probability denotes in a qualitative way all the mechanisms of spalling, cracking, and crunching into dust.

90. The specific values used for failure checking are straightforward. For concrete, only thickness and area are used, both self-explanatory. For wood the quantities for area, length, breaking moment, and period of vibration are required. The area is the outward facing area for rectangular beams and is

$$A = \frac{3}{4} r\ell$$

for round timbers of radius r and length ℓ. Since the maximum stress in a beam of uniform cross section is

$$\sigma = \frac{Mc}{I}$$

where c is the maximum distance from the neutral axis, M is the maximum bending moment, and I is the moment of inertia of the cross section, the maximum bending moment a beam can endure is

$$M = \frac{I\sigma'}{c}$$

where σ' is the stress at failure. The period of vibration of a simple beam of length ℓ is

$$T = \frac{2\ell^2}{\pi}\sqrt{\frac{\gamma A}{EIg}}$$

where γ is the specific weight, A is the cross-sectional area, E Youngs's modulus, and g the acceleration of gravity ($\gamma/g = \rho$ mass density). Timbers in a fortification are not exactly simply supported, nor do they vibrate freely. The effects of stiff supports and damping will act in opposite directions and thus have been ignored. Table 1 gives the equations used in the blast effects calculation along with the quantities used.

Structural Vulnerability

91. The preceding and subsequent sections discuss vulnerability for each discrete component. However, the failure of one component may or may not mean the failure of the structure. In order to describe the interaction with respect to failure between the structural elements, the user must properly link them into a failure system. This linked system models such things as redundant structural elements (posts, beams, etc.) as well as those single elements which must survive in order to continue the mission. Such dependent and independent relationships are modeled with tree structures.

Tree structures

92. The tree structures used in this program are drawn upside down in that the "root" of the tree is at the top of the picture. The convention here is that the system is operational so long as one path from the root to a tip end of a branch remains intact. Some examples of correct, incorrect, and poorly drawn trees are given in Figure 15. The following discussion describes how such trees are interpreted by the user and this program. Note that the numbers in parentheses designate the levels of the tree.

93. Tree (a) shows that seven of eight components (nodes) at level 3 could fail, yet the system would still function provided the levels between the remaining unfailed component and the root were also intact. It also shows that only two level 1 nodes need to fail to incapacitate the system.

94. Tree (b) indicates that the system fails if any one component

Table 1

Equations Used in Blast Effects Calculation

A. Propagation of Blast

$$t_+ = 0.1262 \times 10^{-3} W^{1/3} \lambda^{0.912}$$

$$p = (1 + \cos\theta)\frac{k}{2\lambda^3} \quad \text{(soil)}$$

$$p = \frac{1 + \cos\theta}{2}\left(\frac{4120}{\lambda^3} - \frac{105}{\lambda^2} + \frac{39.5}{\lambda}\right) \text{(air)}$$

where

t_+ = positive phase duration, sec

W = equivalent yield, lb

λ = scaled distance, $ft/lb^{1/3}$

p = peak pressure, psi

θ = angle of obliquity

k = soil constant

B. Breakage

1. Wood:

$$M' = \frac{F\ell}{8}$$

$$M_c = \frac{\left(\frac{\nu s l b d^2}{6}\right)}{12} \quad \text{(rectangular cross section)}$$

$$M_c = \frac{\left(\frac{\nu s l \pi r^3}{4}\right)}{12} \quad \text{(round cross section)}$$

$$P_K = 0.0 \quad \text{if} \quad M < 0.8M_c$$

$$= 0.5 \quad \text{if} \quad 0.8M_c \leq M < 1.2M_c$$

$$= 1.0 \quad \text{if} \quad M \geq 1.2M_c$$

(Continued)

Table 1 (Concluded)

where

M' = applied moment of bending, ft-lb

M_c = critical bending moment, ft-lb

ν = vulnerability factor

s^1 = maximum tensile stress, psi

b = dimension of beam perpendicular to blast, in.

d = dimension of beam along blast, in.

r = radius of log, in.

P_K = probability of collapse for current burst

2. Concrete:

$$P_{ave} = F \frac{1.33}{A}$$

$$\lambda_{EQ} = 3 \frac{10,000}{P_{ave}}$$

where

P_{ave} = equivalent pressure

F = total force

A = area exposed to blast

λ_{EQ} = equivalent distance (scaled)

$$t_1 = 1.051W^{1/3} e^{-0.61\lambda^2} \quad \text{(breach thickness)}$$

$$t_2 = 1.438W^{1/3} e^{-0.434\lambda^2} \quad \text{(heavy damage thickness)}$$

$$t_3 = 1.641W^{1/3} e^{-0.234\lambda^2} \quad \text{(moderate damage thickness)}$$

$$P_K = 0.0 \quad \text{if} \quad t \geq t_3$$

$$= 0.3 \quad \text{if} \quad t_3 > t \geq t_2$$

$$= 0.7 \quad \text{if} \quad t_2 > t \geq t_1$$

$$= 1.0 \quad \text{if} \quad t_1 > t \geq 0$$

where t = scaled thickness

C. Survivor Rule

$$P'_K \leftarrow P_K + P'_K - P_K P'_K$$

where P'_K = cumulative P_K

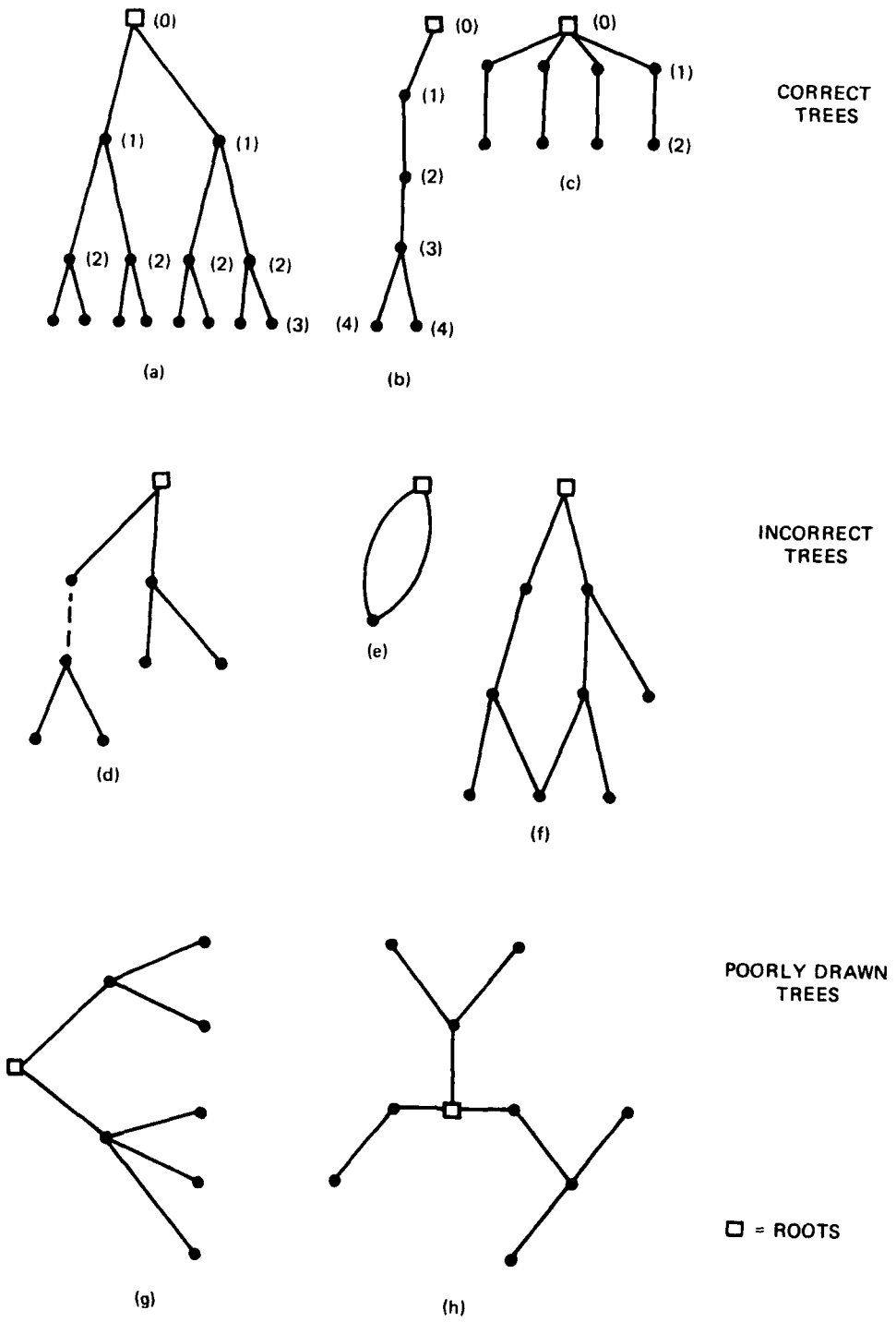


Figure 15. Examples of fault trees

fails at level 1, 2, or 3 or if two fail at level 4.

95. Tree (c) might represent a redundant structural component that could tolerate up to six failures (at levels 1 and 2) of any three branches and still function. However, four failures consisting of one in each branch would incapacitate the system.

96. Tree (d) is not really a tree because it has a missing link indicated by the dashed line. Trees (e) and (f) are also incorrect trees because two branches end at the same tip (each branch must have its own tip).

97. Trees (g) and (h) are poorly drawn but only in the sense that it is rather difficult to identify nodes in different branches that are on the same level. The user is encouraged to use the tree notation described herein in order to reduce definition errors associated with converting from one tree format to another.

Mathematical modeling of trees

98. Trees are defined as graphs which consist of either the null graph or a single node connected by nonredundant edges to one or more independent subtrees, which are themselves trees. As mentioned previously, the trees described in this report have their nodes above their subtrees.

99. In this program, each node lies at a level. The level of a node is defined as the minimum number of edges (i.e., lines between nodes) which must be traversed to reach the root of the tree. The tree is defined as a list of all nodes, the fortification component to which each node corresponds, and level of each node. The physical arrangement in the data implies the interconnection scheme of the tree itself. Each subtree is a continuous sequence of entries delimited at the beginning by the root of the subtree and at the end by another subtree. The next subtree is detected by a node level equal to or less than the level of the subroot. A practical method for describing a tree in this fashion is to proceed recursively performing the following procedure:

a. Write down the root level and label.

b. For each subtree, write down the subtree using this procedure; if no subtrees exist other than null tree, skip this step.

c. Tree definition is complete.

Figure 16 shows an example of using the procedure.

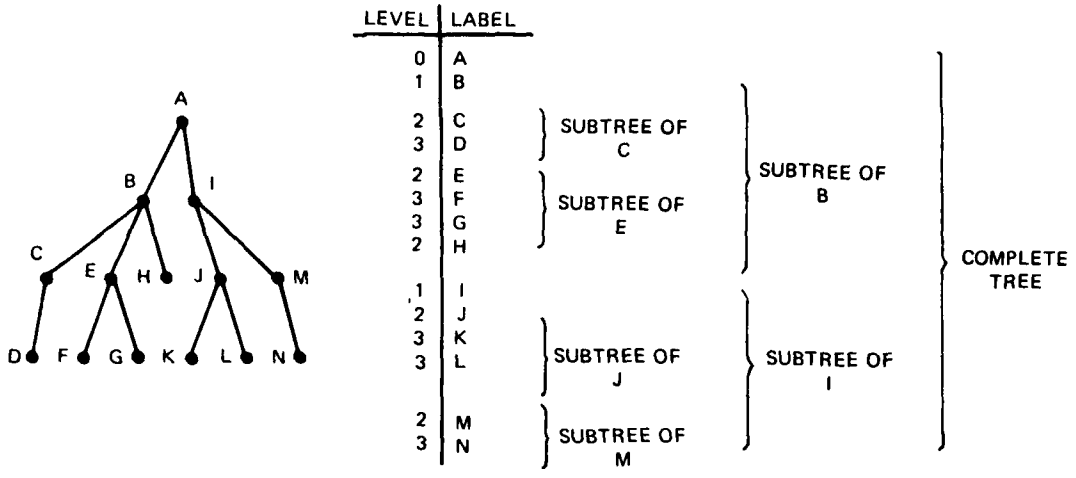


Figure 16. Example of a tree description

Modeling blast-induced structural damage

100. One of the main features of this program is its ability to model the damage produced by both blast and fragments. Blast damage is modeled by actually changing the user's description of the fortification. The program does this by calculating the location and size of the blast-produced crater and then placing a three-dimensional shperical model of that crater in the fortification model. All material previously in the crater is removed. By this means, the cumulative effects of blast-induced damage are approximated. Consequently, successive rounds that fall close to the same point will encounter a more vulnerable model of the fortification.

Mathematical model of the blast crater

101. The program calculates the depth and diameter of a crater caused by a projectile with a given charge. It also computes the

radius and the center of the approximating sphere. The equations used to determine crater diameter and depth are fits to the curves given in Figure 3B1A, page 422, of "Terminal Ballistics." These two empirical equations have the same form

$$\text{diam, depth} = A - B|X - C|^{D} + E|X|^{F}$$

where

A,B,C,D,E,F = empirical constants dependent on the soil type, and whether the equation is being used to determine crater diameter or depth

X = depth of the charge center of gravity below the surface at burst divided by the cube root of the charge weight

diam = crater diameter

depth = crater depth

102. The equations used to determine the radius and center point of the approximating sphere are determined form the geometry to be

$$R = \left(\frac{1}{8}\frac{\text{diam}^2}{\text{depth}} + \frac{\text{depth}}{2}\right)$$

$$Z_c = R - \text{depth}$$

where

R = radius of approximating sphere

Z_c = Z coordinate at center of approximating sphere

103. The resulting crater is modeled with a 24-sided, 38-point triangular approximation of a sphere. The sphere is used to portray damage due to cratering by adding it to the target description as a damage component.

Mathematical model of the sphere

104. The coordinates of the triangular approximation are the six intersection points of the sphere with an X_s, Y_s, Z_s axis system (located at the center of the sphere) and one point in each of the eight quadrants. In order to make all triangles of equal size, the coordinates of these

latter points must be equal and lie on the sphere. In Figure 17, the center of the sphere is located at X'Y'Z'. A second coordinate system X_s,Y_s,Z_s is established with its orgin at X'Y'Z'. The aforementioned quadrant point Q is located at X,Y,Z in the X_s,Y_s,Z_s system. Considering the point Q in the first quadrant (X, Y, and Z are positive), find values for the aximuth θ and elevation ϕ angles that produce the aforementioned equal size triangles. The equal-size condition is met when the normal distances from Q to X_s, Y_s, and Z_s are equal. The equal-distance (hense equal-size) condition is met when $X = Y = Z$. The polar equations in Figure 17 express the coordinates of a point on a sphere. Solving them for θ when $x = y$ yields

$$R \sin \phi \cos \theta = R \sin \phi \sin \theta$$

$$1 = \frac{\sin \theta}{\cos \theta} = \tan \theta$$

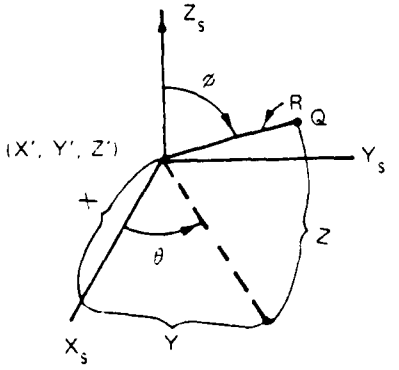
$$\theta = \arctan 1 = 45^{o}$$

Substituting 45^{o} for θ , setting $X = Z$ and solving for ϕ yields

$$R \sin \phi \cos 45^{o} = R \cos \phi$$

$$\frac{\sin \phi}{\cos \phi} = \frac{1}{\cos 45^{o}} = 1.414 = \tan \phi$$

$$\phi = \arctan 1.414 = 54.7^{o}$$



$X = R \text{ SIN } \phi \text{ COS } \theta$
$Y = R \text{ SIN } \phi \text{ SIN } \theta$
$Z = R \text{ COS } \phi$

Figure 17. Sphere's quadrant point

Therefore,

$$X = 0.5773R$$

$$Y = 0.5773R$$

$$Z = 0.5773R$$

Simulating fragment damage

105. A unique feature of this program is its ability to model the damage-producing and lethal potential of fragments produced by airburst (proximity-fused) weapons. The program does not, however, model fragment damage from contact- or delay-fused weapons. The assumption in the latter case is that the blast component of the weapon will be the dominant agent in producing damage. The program further assumes that any burst initiated inside of the fortification (i.e., when the structure is perforated) will kill all components and personnel therein.

Number of fragments impacting components

106. The components to be tested for exposure to fragment damage are lined up on rays in order of increasing distance from the burst point. The rays from the burst point thus define the straight-line paths over which the fragments will pass.

107. For each ray, the program performs the following steps. The program decides which nose band of fragments the ray lie in. The fragment velocity pairs associated with that band are sent down the ray toward the first component. The actual number N of fragments of each weight sent toward the component is

$$N = N_B \frac{A_G}{A_B}$$

where N_B is the user-specified number of fragments of that weight group in the band through which the ray passes, A_B is the area of a unit sphere covered by that band, and A_G is the area of a unit sphere covered by the grid element associated with the ray.

Modeling the air
drag on the fragments

108. During the motion of the fragment from the burst point to the first component on each ray, the fragment is assumed to be moving through air. If the distance moved is less than 2 ft, the air drag is assumed to have no effect on the fragment velocity. In this case the impact velocity V_I will be the starting velocity V_S given in the fragment data supplied by the user. If the distance from the burst point to the first component is 2 ft or more, the program uses the following equation to calculate the impact velocity:

$$V_I = V_0 \, e^{-(0.002363 \, D/W^{1/3})}$$

where

D = distance from the burst point to the first component on the ray, in.

W = weight of the fragment, grains

Fragment penetration/perforation

109. The program calculates the depth of penetration in the first component and the volume of material removed by that penetration. The volume of material removed is the quantity within a truncated cone whose height is the depth of penetration and whose apex angle depends on the material being penetrated (5, 60, and 90 degrees for wood, sand, and concrete, respectively). If the component is perforated (i.e., if potential perforation is greater than the line-of-ray thickness), then the volume removed is that of a twice truncated cone whose height is the line-of-ray thickness (see Figure 18).

Exit velocity

110. When the potential for penetration E into a massive amount of material is greater than the thickness T of the component, then the fragment exists that component at a velocity less than the entrance velocity V_E. For all component materials except personnel, the exit velocity V_X is calculated from the formula

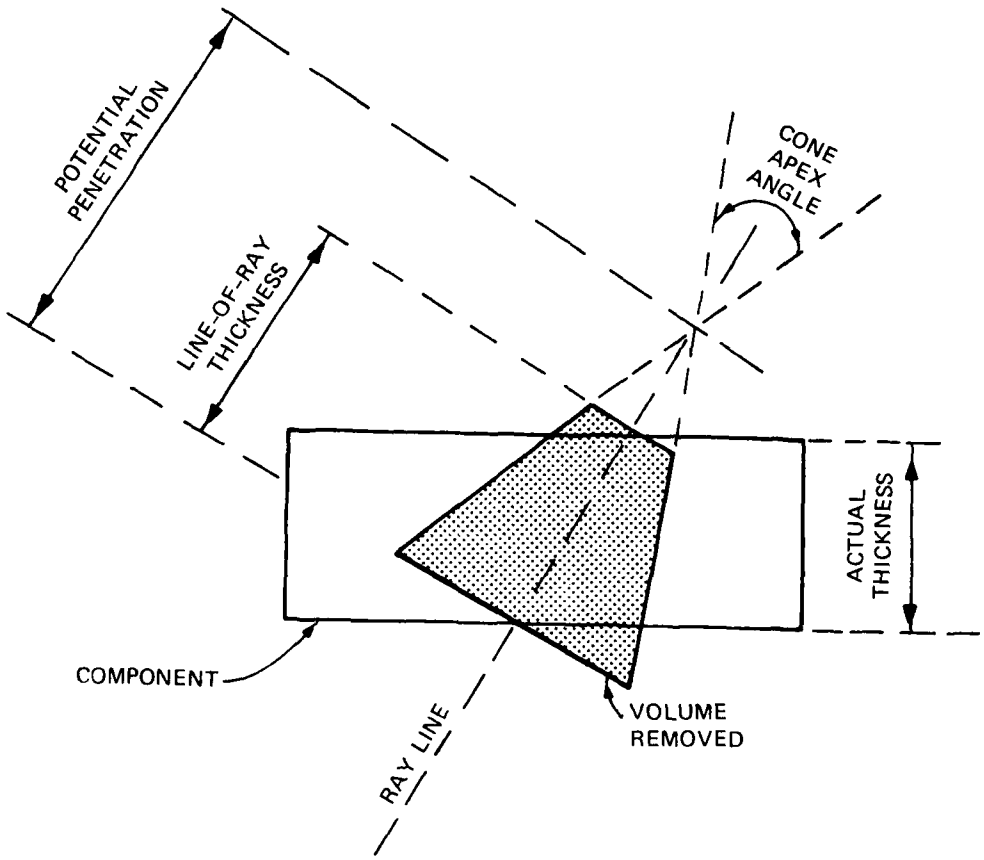


Figure 18. Volume removed by perforating fragments

$$V_X = V_E(1 - T/E)^{1.8}$$

For personnel, the formula is taken from equations of the form

$$V_x = V_E - K_1 T W^{K_2} V_E^{K_3}$$

where

K_1, K_2, K_3 = empirical constants
T = line-of-ray thickness, in.
W = fragment weight, grains
V_E = entrance velocity, ft/sec
V_x = exit velocity in ft/sec

111. Fragments which perforate the first component are passed on to the successive components on the ray. Each successive impact occurs at the exit velocity calculated for the previous component on the ray. This process continues until the fragment fails to perforate a component. Note that the weight of the fragment is not reduced as it passes through components. This procedure is applied to all combinations of fragment weight and velocity specified for this nose band.

Damage assessment

112. The program repeats the above calculations for every ray from the burst point. When all rays have been used, the program divides the cumulative volume removed from each component by the initial volume of the component to produce a new effective density value for each component. (The effective density for each component is set equal to one at the beginning of each attack.) For subsequent bursts, the line-of-ray thickness of each component is multiplied by the effective density value so as to reduce the apparent thickness and thereby simulate the damage done by removed material. The effect of this reduced apparent thickness is to make it easier for fragments to perforate successive components and eventually gain access to components protected by the structure.

Penetration Equations

Fragment shape

113. The penetration equations for sand, concrete, and soil are tuned to calculate the penetration of a rectangular solid fragment whose dimensions are $\ell \times \ell \times 5\ell$. The fragment is assumed to impact on the small face $\ell \times \ell$ with its long axis parallel to its line of motion.

Sand

114. The equation for penetration into sand from the Poncelet equation (also used to calculate projectile penetration into soil) has the form

$$D_p = \frac{m}{2Ab} \ln \left(1 + \frac{bV^2}{a}\right)$$

where

D_p = penetration distance, ft

m = fragment mass, slugs

A = fragment presented area, ft^2

V = fragment impact velocity, ft/sec

a = empirical constant, lb/ft^2

b = empirical constant, $lb\text{-}sec^2/ft^4$

115. The constant terms in this equation were adjusted to match the curves for fragment penetration through sand in an Air Force Weapons Laboratory (AFWL) report* (pages 5-27). The resulting expression for penetration E in inches for fragments of weight W in grains moving at velocity V in feet per second is

$$E = 0.349536W^{1/3} \ln (1 + 2.88 \times 10^{-5}V^2)$$

Concrete

116. The penetration equation for concrete, taken from a paper by A. K. Kar,** is

$$P = \frac{a}{\left(f'_c\right)^{1/2}} N_2 \left(\frac{E}{E_n}\right)^{5/4} \frac{W}{D} \left(\frac{V}{d}\right)^{9/5} + 1$$

$$E = 2 (P - 1)^{1/2} \quad \text{for} \quad P < 2$$

$$E = Pd \quad \text{for} \quad P \geq 2$$

---

* R. E. Crawford et al., "Protection from Nonnuclear Weapons," TR-70-20-120, February 1971, Air Force Weapons Laboratory, Kirtland Air Force Base, Albuquerque, N. Mex.

** A. K. Kar, "Local Effects of Tornado-Generated Missiles," Journal of The Structural Division, American Society of Civil Engineers, Vol 104, No. ST-5, May 1978, pp 809-816.

where

W = projectile weight, grains

V = projectile impact velocity, ft/sec

N_2 = nose shape factor (0.72 for fragments)

D = outside diameter of nonsolid nose shapes, in.

d = diameter of a circle whose area equals that of the impact face, in.

E_m = elastic modulus of mild steel

E = elastic modulus of the fragment

f'_c = compressive strength of concrete, psi

a = $0.10237 \quad 10^{-6}$

P = penetration in calibers E/d

E = penetration, in.

117. The program value for E/EM is one; f'_c is 5700 psi; and D equals d for the solid fragment face. The effective diameter is

$$d = \frac{2W^{1/3}}{\pi^{1/2} 5^{1/3} \gamma_F^{1/3}}$$

where γ_F is the unit weight of the fragment. Using 490 lb/ft^3 = 1984.95 $grains/in.^2$ for the unit weight, the resulting expression is $d = 0.05250674W^{1/3}$.

Wood

118. The penetration equation for wood is an empirical relationship based on data reported elsewhere (classified reference*).

$$E = V^c W^d (\sec\phi)^e \eta^f$$

where

E = penetration, in.

V = fragment impact velocity, ft/sec

W = fragment weight, grains

θ = obliquity angle measured from the surface normal

---

* Bibliographic material will be furnished to qualified agencies upon request.

η = Young's Modulus for the impacted wood (10^6 psi)

c,d,e,f = empirical constants

The resulting equation for penetration in inches is

$$E = 7.031976 \times 10^{-8} V^{2.15068} \times W^{0.29279} \times \sec\theta^{-0.78629}$$

This equation yields low values for penetration, indicating that the wood modeled is extremely hard. The leading coefficient should be adjusted to match penetration data from tests on the variety wood to be used. Other alternatives are noted in Appendix A.

Ricochet Modeling

119. When this program was first developed, ricochet was not modeled. All fragments were treated with respect to penetration as if their line of motion at impact was normal to the surface. The apparent thickness T was increased over the actual thickness T_A with

$$T = \frac{T_A}{\text{cosin } \theta}$$

where θ is the angle of obliquity (i.e., the angle between the surface normal and the fragment line of motion; e.g., θ is 90 degrees when the line of motion is parallel to the surface). This increased thickness correctly modeled the longer penetration path, but caused excessive volume to be removed since the removal equations are also based on normal impacts. (See Figure 18 and note the large cone removed for oblique impact.)

120. This generated unrealistic damage from fragments striking the surface at highly oblique angles (i.e., $\theta > 40$ degrees). To overcome this difficulty, a ricochet model was developed for concrete from the graphical data presented on pages 5-11 of AFWL TR-70-127. The formula used is

$$\theta_R = \tan^{-1} CV$$

where θ_R is the limit angle of obliquity for ricochet, V is the impact velocity, and C is an empirical constant which has the value 0.00041666667 in this program. This expression closely follows the graphical representation giving θ_R = 45 degrees for V = 2400 ft/sec and 22 degrees for V = 1000 ft/sec . At higher velocities (for which no data were found), this expression may not be valid (e.g., θ_R = 63 degrees at 5000 ft/sec and 76 degrees at 10,000 ft/sec).

121. The program compares the angle of obliquity ϕ of the fragment line of motion at the surface with the ricochet limit angle θ_R . If ϕ is greater than θ_R , the fragment is assumed to have ricocheted from the surface. No damage is accessed for that fragment. Neither is its path traced to other surfaces that it might impact. Ricochet models have not yet been incorporated for soil, sand, wood, metal, etc. These and their related limitations are discussed in Appendix A.

Component Vulnerability to Fragments

122. As is the case for blast, the assessment of damage due to fragments is a relative one. It is stated in terms of the likelihood (i.e., probability) that failure has occurred. In order to calculate the probability of kill after being hit $P_{k/h}$ by fragments, the program must relate $P_{k/h}$ to the weight and velocity of the fragments that hit each protected component.

123. The program has two ways to express this relationship. For personnel, the program uses a built-in empirical function (Kokinakis equations). This function determines $P_{k/h}$ from the fragment weight and velocity for major portions of the human body. It can, where such detail is necessary, model the different $P_{k/h}$ for hits in an arm versus the torso.

124. The other protected contents are modeled through user-supplied data in the form of piece-wise curves that relate $P_{k/h}$ to velocity for a specific fragment weight. Given a particular fragment weight and velocity, the program interpolates a $P_{k/h}$ from the

appropriate curves for the component hit.

125. This graphical modeling approach is quite flexible. It allows data based on field tests to be used directly instead of forcing the user to develop an algebraic model for vulnerability. However, along with that flexibility comes some added complexity and associated questions on how the program handles specific situations. The following paragraphs set out the program action for all cases.

User-Supplied Data

126. The user supplies the following data for each different group of critical components. The following discussion applies to these data for such groups:

a. SMALL. This is the weight in grains of the smallest fragment that can damage the component.

b. W^k. This is the fragment weight in grains to which the weight curve k applies. The superscript k indicates there are several weight curves (i.e., $k = 1$, $k = 2$, $k = 3$, ... to $k = M$, the last curve).

c. VEL_J^k and P_J^k. These are points on the weight curve k which relate velocity VEL to probability of kill P. The J subscript indicates that there are several points (i.e., $J = 1$, $J = 2$, $J = 3$, ... to $J = n$, the last point). The maximum value of N is 8.

127. The above data are used in developing a computer-based equivalent of drawing several curves on one graph. Each curve represents velocity versus $P_{k/h}$ for one weight. What follows is a description of how the program selects which curve or pair of curves to use; then, how it interpolates on and between those curves to calculate $P_{k/h}$ for a specific fragment weight W_T moving at a given velocity V.

128. The fragment weight W_T is compared to SMALL. If $W_T <$ SMALL then W_T is too small to harm the component. Therefore, $P_{k/h} = 0$. If $W_T \geq$ SMALL it may still be less than the weight associated with the first fragment curve W^1. In this case, the program then interpolates in weight curve 1 for the probability P associated with velocity V using the linear interpolating formula

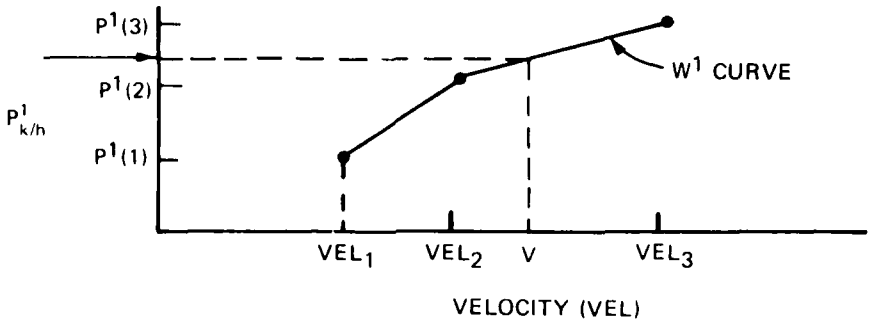


Figure 19. $P_{k/h}$ computation for $WT < W^1$

$$P^1_{k/h} = \frac{V - VEL^1_{J-1}}{VEL^1_J - VEL^1_{J-1}} \left(P^1_J - P^1_{J-1}\right) + P^1_{J-1}$$

which is illustrated in Figure 19. The above conditions apply when V lies between any pair of velocity values of the weight curve. In the case where $V < VEL^1_1$, $P^1_{k/h} = 0$, when $V > VEL^1_N$, $P^1_{k/h} = P^1_N$. The user should keep this in mind when preparing the curves to be sure that $P_{k/h}$ values obtained for underrange and overrange conditions are reasonable.

129. The $P^1_{k/h}$ computed above is for a fragment weight of W^1. However, as mentioned previously, the actual fragment weight W_T was assumed to be less than W^1. Therefore, the $P_{k/h}$ for the W_T weight is obtained by using the adjusting factor A and the $P^1_{k/h}$ value with the formula

$$P_{k/h} = P^1_{k/h} A$$

where

$$A = \frac{W_T - SMALL}{W^1 - SMALL}$$

130. In the case where W_T is between two of the weight curves, W^k and W^{k-1}, the $P_{k/h}$ adjusting factor is

$$A = \frac{W_T - W^{k-1}}{W^k - W^{k-1}}$$

The probability at V on the lower curve for weight W^{k-1} is

$$P_L = \frac{V - VEL_{J-1}^{k-1}}{VEL_J^{k-1} - VEL_{J-1}^{k-1}} \left(P_J^{k-1} - P_{J-1}^{k-1}\right) + P_{J-1}^{k-1}$$

The probability at V on the upper curve for weight W^k is

$$P_U = \frac{V - VEL_{\ell-1}^{k}}{VEL_{\ell}^{k} - VEL_{\ell-1}^{k}} \left(P_{\ell}^{k} - P_{\ell-1}^{k}\right) + P_{\ell-1}^{k}$$

The probability is then linearly interpolated between the upper and lower bounds by using the adjusting factor in the equation

$$P_{k/h} = (P_U - P_L)\ A + P_L$$

This procedure is illustrated in Figure 20.

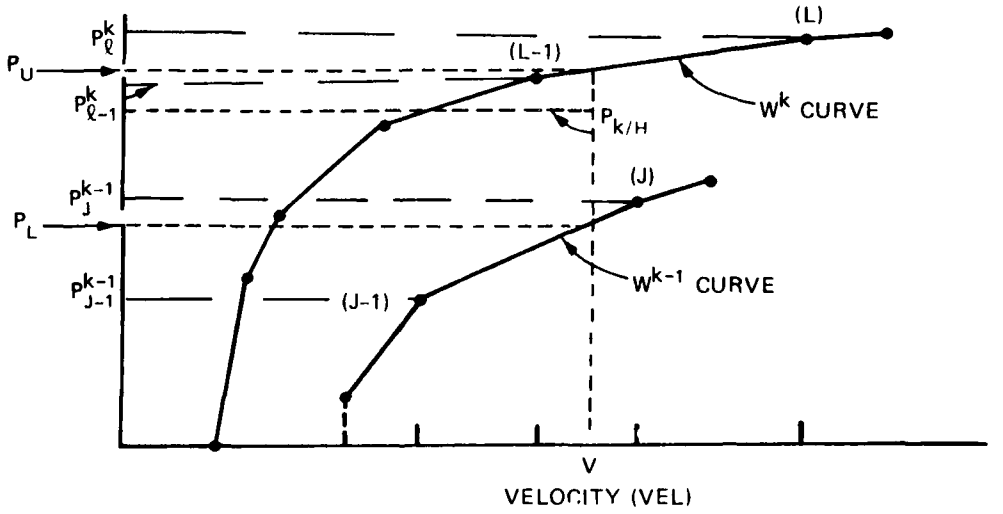


Figure 20. $P_{k/h}$ interpolation

131. Note that if V lies beyond the first or last value for either or both curves, the value of P_U or P_L will be set to the probability values for the first or last point in the respective curves (see paragraph 129).

PART III: CUMULATIVE DAMAGE AND SURVIVAL STATISTICS

Status at this Point

132. The steps and procedures described thus far have simulated blast and/or fragment damage from the burst of one projectile. The damage produced is modeled by:

a. Changes in the fortification's physical description by incorporating the blast-induced crater.

b. Reductions in the effective density of components that have been hit by fragments.

c. Reductions in the survival probabilities $P_{s/h}$ of critical components that have been damaged by blast and/or fragments.

Continuing the Attack

133. The program repeats the attack process by sending another projectile toward the previously established aim point. As mentioned previously, the fortification will be subjected to damage produced by a series of projectiles that burst at different points around the aim point. The attack is complete when the fortification has been exposed to assault by the user-specified number of shots in one attack.

The Attack Summary

134. Upon completing the attack, the damage indicators are in the probability of survival statistics. The crater modeling and effective density reductions have let each successive projectile take advantage of the damage done by those which preceded it, thereby having greater likelihood of defeating a critical component.

135. At the end of each attack, the survival probabilities for both blast-induced $P^{B}_{s/a}$ and fragment-induced $P^{F}_{s/a}$ damage are used to calculate a survival probability for the fortification via the formula

$$P_{s/a} = P^{B}_{s/a} \; P^{F}_{s/a}$$

136. This is the survival probability for one attack. It is vital to understand that very little meaning with respect to hardness can be attached to this number. Since this process is a statistical one, many such measurements must be made in order to justify using the concepts of statistics and probability. Consequently, many attacks must be made on the as-new fortification before any statement can be made about $P_{s/a}$ for a specific aim point, and error ellipse.

Repeated Attacks

137. The program begins each new attack by setting the fortification back to its initial state, the effective densities to one, and all P_s to one. It then repeats the attack process and calculates a new $P_{s/a}$. Note that, since the burst points change in a pseudorandom way, the $P_{s/a}$ for each attack will be different. The program saves those individual $P_{s/a}$ values for use in the hardness evaluation. The attacks will continue to be made until the number of attacks specified by the user has been reached. Since there is no way to restart this program, all attacks must be made in one continuous run on the computer.

Hardness Evaluation

138. When all of the attacks have been completed, the program uses the N-attack statistics to calculate $P_{s/k}$, the average probability of surviving k shots for $k = 1, 2, 3, \ldots M$ (where M is the number of shots per attack). In addition, it produces the standard deviation of $P_{s/k}$, the standard error of $P_{s/k}$, an attack summary, and a table containing the average $P_{s/k}$ and the standard deviation of $P_{s/k}$. The program then uses the various plotting subroutines to generate a single graph containing the average $P_{s/k}$, the upper and lower 95 percent confidence limits of the standard deviation, and the upper and lower 95 percent confidence limits of the standard error, all as a function of the number of shots.

Statistical Formulas Used

139. For N data points, the mean is

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N}$$

The standard deviation is

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} X_i^2}{N - \bar{X}^2}}$$

and the standard error is

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{N}}$$

Interpretation

140. These data give the user a measure of hardness in terms of $P_{s/k}$ and a measure of how much confidence may be placed in that survival probability. The graphs show whether or not the procedure appears to be converging toward the average probability curve. This graph is particularly important because some complex fortifications may require simulating a large number of attacks in order to produce a convergence.

PART IV: CONCLUSIONS AND RECOMMENDATIONS

A Useful Tool

141. Throughout Parts I through III of this report and in Appendix A, the author has noted the program's approximations, limitations, etc. Those limitations have perhaps been overstressed in an effort to describe exactly what is and is not being modeled. In spite of these limitations, the author believes this program is an effective adjunct to hardness estimation.

A Step in the Right Direction

142. This effort is a first step. It needs to be used, critically evaluated, and improved. It cannot be used alone.

An Adjunct to Field Testing

143. Verification by field testing is a vital element in hardness evaluation. Only because of the high cost of extensive field testing is there any need for efforts like this. But high cost is a reality; therefore, this program can best be used to help make the most cost-effective use of field testing by:

a. Clearly showing that some fortifications are either excellent or inadequate. Such statements can be made about fortifications which are only marginally influenced by the limitations of this program.

b. Pointing out vulnerable structural parts that should be probed during the field testing. It is noted that good engineering judgment often identifies these points before the test. However, such good engineering judgment comes, by and large, from direct or indirect experience. Thus, we tend to stick with proven techniques, largely because it is so difficult and expensive to gain experience in new approaches. This program can provide a low-cost way to gain some new experience.

c. Adding a new dimension to hardness evaluation through statements about relative hardness.

Immediate Needs

144. The immediate needs at this point are twofold. On the one hand, this program needs to be used. It needs to be used in order to show potential users what a probabilistic approach to hardness evaluation means to them. On the other hand, it needs to be used to show its developers where to concentrate their enhancement efforts. The areas needing improvement are many, but which are the important ones? This critical question needs an accurate answer.

Future Direction

145. The work done in this hardness evaluation effort is taxing the capability of current computers to produce cost-effective answers. These engineering research analysis tools, like dynamic finite element analysis, nonlinear material modeling, and other many-body problems, are also pushing the computer's capability to produce cost-effective answers. Thus, today the size and complexity of the problems we solve and the accuracy with which we model them is largely dependent on the cost-effectiveness of computers available to us.

146. Developing the capability to accurately simulate every facet of weapon-fortification interaction is an extremely complex task. If we had that capability today, we would probably find it too expensive to use. However, better computer capability is perhaps closer to reality than we realize. More important is the fact that the cost of other hardness evaluating methods is rising while the cost of computer-based analysis continues to decline. Therefore, we are being driven to use computer-based procedures by the very same forces that drove our predecessors to physical modeling. We are driven to computer-based analysis by the unbearable burdens of time, money, and victory versus defeat that go with solving problems any other way.

147. Consequently, it is vital that we develop and maintain an advanced capability in computer-based hardness assessment. Victory in future conflicts may well go to the nation with superior decisionmaking

capability. Advanced simulation capability may become a key element in making high-quality decisions on hardness. It is difficult to imagine how an analyst of the future, handicapped by inferior computer procedures, can be expected to produce a winning system of fortifications or a winning attack strategy.

148. Thus, the effort of the future must include research, development, testing, and evaluation in the multitude of computer-based capabilities needed to accurately evaluate hardness. We expect the use of this program will expose and explore some of these areas.

PART V: PREPARATION OF INPUT

Organization

149. The first section of this part contains a short description of terms used in the instructions; terms with which the occasional computer user may not be familiar. The remainder deals with the data and format requirements of each card group used in the program. The short explanations given are directed to data preparation and definition requirements. These are supplemented by paragraph references to the places in this report where the data are more fully discussed.

Definition of Data Preparation Terms

150. Two kinds of data are required by this program. They are fortification geometry data and parameter data. In the presentation of the parameter data requirements, several column headings are used. They should be interpreted as follows:

a. COLUMNS. This entry gives the location and indirectly the width of the data field (i.e., the number of card columns set aside for the entry). It specifies the numbers of the first and last columns of the card that must contain the data value. Note that the data field specification is inclusive. For example, the entry "6-10" means the data field is 5 columns wide; it consists of columns 6, 7, 8, 9, and 10 which are available for recording the value. The data value must be right-justified in the data field (i.e., there must not be any blank columns between the rightmost digit of the value and the right end of the data field).

b. FORMAT. This entry is taken from the FORTRAN programming language; consequently, some users may be unfamiliar with it. It is used to specify the kind of data allowed, the width of the data field, and where the decimal point is assumed to lie. The three specifications used in this report are:

(1) Integer specification (In). This is indicated where the letter given is an "I." Following that letter will be a number (n) which specifies the data field width. Decimal points must not be given with this

type of input. The decimal point is assumed to lie immediately to the right of the data field. Should the user inadvertently leave blank columns (Ø) between the last digit and the end of the data field, those blanks will be treated as if they were zeros (e.g., Ø3ØØØ becomes 3000). It is bad practice to use this feature. If "3000" is intended, it should be recorded as Ø3000. If "3" is intended, it should be recorded as ØØØØ3 in the data field (here assumed to be 5 columns wide).

(2) Real number specification (Fn.m). This is used for data values which may have fractional parts and is indicated by the letter "F." The number n is the field width. The number m indicates that the entry is assumed to contain m fractional digits unless a decimal point is supplied with the data. For example, "F12.2" means that 12 columns are reserved for the real data value. The last 2 columns are reserved for the tenths and hundredths values. Thus, the right-justified entry "3472" is interpreted as 34.72. However, the entry "396.1722" (i.e., decimal supplied) overrides the assumed decimal position and will be used as specified.

(3) Alphabetic and numberic specification (An). This is used to indicate via the "A" that letters and/or numbers are supplied in a data field that is n columns wide. This is typically used for names, labels, titles, etc.

c. UNITS. This entry specifies the physical units of the data value (e.g., feet, inches, etc.).

d. DESCRIPTION. This entry gives a short description of the data item and/or how it is used in the program.

Recording Values

151. The recorded values must be right-justified in the data field. Negative values are indicated by a negative sign to the left of the leading (leftmost) digit. Values larger or smaller than the data field width allows may be input via the exponential specification. For example, the number -1,600,000,000 (i.e., -1.6×10^9) cannot be recorded directly under a F10.2 specification. The user may record the value as "-1.6E9," but it must be right-justified in the field. The small negative number -6×10^{-14} would be recorded as "-6.E-14" in the data field.

Input Data Card Groups

Card group 1; optional output and grid size

152. The first card group consists of just one card. It is used to turn on the optional output (a program testing feature) and establish the size of the large and small grids.

Group 1 Format

Columns	Format	Units	Description
1-5	I5	none	Output from subroutine PB4*
6-10	I5	none	List of bounds of components*
11-15	I5	none	Output of subroutine PSORT*
16-20	I5	none	List of rays in the ray file*
21-25	I5	none	Number of small grid elements in the azimuth circle (see paragraphs 64-66)
26-30	I5	none	Number of large grid elements in the azimuth circle (see paragraphs 64-66)
31-35	I5	none	Deletes the survivability plot when this value is one

* Supply zero (or leave blank) to delete the output. Output is produced when the value is one.

Card group 2; glitter spheres

153. The cards in this group each define a glitter sphere (see paragraph 71). The maximum number of glitter spheres is 10. Place a blank card after the last glitter sphere card to indicate the end of this group. If there are no glitter sphere data, supply one and only one blank card.

Group 2 Format

Columns	Format	Units	Description
1-10	F10.0	in.	X coordinate of the glitter point
11-20	F10.0	in.	Y coordinate of the glitter point
21-30	F10.0	in.	Z coordinate of the glitter point
31-40	F10.0	in.	Radius of the glitter sphere

Card group 3;
component specifications

154. Each component of the fortification has a code number which refers to the component code numbers in this group of data. Associated with each code number are several other code and specification data that identify the physical characteristics of that component and specify how it is to be modeled:

a. Code number. Each component should have a unique code number and associated set of data in this group. Note that a unique value of effective density is associated with each component code number. Thus, assigning the same code number to two components indicates that damage to one also damages the other (a highly unusual situation). The maximum number of component code numbers is 200.

b. Structural parameter data. The structural parameter number refers to the parameter data in card group 7. It is vital that all components which make up the structure (posts, beams, slabs, etc.) and those whose degradation (sandbags, etc.) will expose the structure to greater damage be properly modeled with these data. In this regard, soil several feet away from the base of the structure might not be considered critical, but sandbag protection would be critical and should be modeled with the structural parameter data.

c. Damage characteristic number. The damage characteristics number identifies where the program accumulates the total force and minimum pulse time incident on a component. Where several components act together in their response to force and pulse time, this situation can be modeled by assigning the same damage characteristics number to each component. Normally, each component will have a unique damage characteristic number so that the forces acting on it are due only to the forces on that component. The numbers assigned should begin with 1 and increase in steps of 1. The maximum value for the damage characteristics number is 100.

d. Data sets. Note that up to five sets (S = 1, 2, 3, 4, 5) of component code numbers and associated data may be recorded on one card. The card columns shown below are for set 1 (S = 1). The card columns for the other sets are easily obtained by adding 15 (S - 1) to the set 1 column numbers. The ability to place several sets of data on one card is useful in compressing data. However, some users find it difficult to locate and change data in compressed form. These users may prefer to record only one set of data per card.

e. End of data indicator. The end of this group of data is indicated by placing a blank card after the last data card in the group.

Group 3 Format

Columns	Format	Units	Description
1-5	I5	none	Component number
6-7	I2	none	Structural parameter number (see paragraph 160)
8-9	I2	none	Damage characteristics number (see subparagraph 153c)
10	I1	none	Component type; use 1 for a critical structure, 2 for protected contents, and 0 for all other components
11	I1	none	Ray density; use 1 to model the component with high-density days (i.e., small grid rays), 2 for low-density modeling (i.e., large grid rays), and 0 when the ray density is not critical
12-13	I2	none	Material code numbers (see paragraph 155)
14-15	I2	none	

155. This program provides for several different materials. The material is specified by giving the proper material number in the component specification data. The following numbers are used:

Material Number	Kind of Material
1	Concrete
2	Wood
3	Water
4	Arms
5	Leg
6-10	Not used
11	Sand
12	Sandy loam
13	Loam
14	Clay
15	Rock

Card group 4; shot control

156. This data group consists of a single card. It contains several data items not previously discussed:

a. Print control. The first item is a switch which is used to turn on the optional printed output. This output consists of a list of the components that are the same material (i.e., the components by order of material numbers). This list is useful in checking the data to see that every component is correctly defined.

b. Projectile entry step distance. The second item is the initial distances in inches that the projectile will cover in its movement along the straight-line entry path from its starting point at the highest elevation of the fortification toward the aim point (see paragraphs 39 and 40).

c. Random number starting seed. This item is the start-up (seed) value for the random number generator. As mentioned previously, the generated numbers are not truly random, a fact alluded to when referring to a chain of random numbers. The implication here is that the end of the chain is linked to the beginning (i.e., an endless chain). Continuing with the analogy, we may "begin" at any point on an endless chain. In like manner, the seed number selects the starting point in the program chain of random numbers. This value may be changed and attacks repeated to see what, is any, influence the starting point on the random chain has on the final results.

d. Hit/miss box dimensions. The last group of items defines a hit/miss box around the structure and its protective soil. This box is used by the program to decide whether a shot has missed the target. The box is a rectangular solid whose sides, top, and bottom are planes which are parallel to the X,Y, Y,Z, or X,Z planes. The user defines the box by giving minimum and maximum values of X, Y, and Z which in turn define the position of the near and far side X, the right and left side y, and the top and bottom Z. The user should try to define a box that is just large enough to define the potential damage space around the structure and the protecting soil. Note that this specification is not an extremely critical one in that the box need not be the smallest possible one. However, it should be only slightly larger than the minimum size in order to save computer time that would be wasted on needless burst calculations that cause no damage.

Group 4 Format

Columns	Format	Units	Description
1	I1	none	Print switch for components versus material; 0 means no print, 1 means print
2-5	I4	in.	Step size on projectile entry path
6-15	I10	none	Random seed value
21-30	F10.0	in.	Minimum X value for the box face
31-40	F10.0	in.	Maximum X value for the box face
41-50	F10.0	in.	Minimum Y value for the box face
51-60	F10.0	in.	Maximum Y value for the box face
61-70	F10.0	in.	Minimum Z value for the box face
71-80	F10.0	in.	Maximum Z value for the box face

Card group 5; attack data

157. There is only one card in this group. It defines the error ellipse, the aim point, the direction of the entry path from which the projectile is delivered (see paragraph 39), the number of shots per attack, and the number of attacks. Note that the error ellipse is in the X, Y plane and its center is at the aim point. Its major axis is parallel to the entry path azimuth. The distance along the major axis from the aim point to the ellipse is the "range error" required in this data group. The like measured distance along the minor axis is the "deflection error" required in this data group. Note that the range error and the deflection error referred to above are, respectively, the standard deviations for range and deflection (i.e., σ_R and σ_D). For the special case where $\sigma_R = \sigma_D$, the error ellipse becomes a circle. In this case, about 39 percent of the shots will fall inside the error circle. Frequently, the weapon precision is given in terms of the circular error probable (CEP). The CEP refers to the radius of a circle in which about 50 percent (not 39 percent) of the shots will fall. In order to use this CEP specification, the σ_R and σ_D values supplied to the program must be set to CEP/1.1774 For the case where the weapon precision is given in terms of range error probable (REP) and deflection error probable (DEP), the following approximation may be used to

convert those values for use in this program:

$$\sigma_R = \frac{REP}{0.674}$$

$$\sigma_D = \frac{DEP}{0.674}$$

Group 5 Format

Columns	Format	Units	Description
1-10	F10.0	in.	Range error
11-20	F10.0	in.	Deflection error
21-30	F10.0	in.	Aim point X coordinate
31-40	F10.0	in.	Aim point Y coordinate
41-50	F10.0	degrees	Entry path azimuth
51-60	F10.0	degrees	Entry path elevation (must be greater than zero)
61-65	I5	shots	Number of shots per attack
66-70	I5	attacks	Number of attacks

Card group 6; projectile and fusing

158. This data group consists of only one card. It specifies the overall weight, the velocity the projectile has at impact (or burst for proximity-fusing), the diameter of the projectile, the fuse delay/burst distance, the weight of the charge, and the nose code.

a. Fusing. The fuse specification has two allowable forms. It may specify the time in seconds that will elapse between impact with the first component and burst initiation. In the other case, a negative sign is used to indicate proximity-fusing. The value specified (sign ignored) is the distance in inches at which the burst will be initiated. (See paragraph 44 for a discussion of how this distance is measured.)

b. Nose type number. The nose type number is used to specify the shape of the projectile nose. The code is 1 for blunt, 2 for average, and 3 for sharp projectile noses.

c. Velocity. The velocity specification must be greater than 10 ft/sec. This is discussed in paragraph 45.

Group 6 Format

Columns	Format	Units	Description
1-10	F10.0	lb	Projectile weight
11-20	F10.0	ft/sec	Projectile velocity
21-30	F10.0	in.	Projectile diameter
31-40	F10.0	seconds or in.	Fuse delay time or proximity distance
41-50	F10.0	lb	Charge weight (TNT equivalent)
51-55	I5	none	Nose type number

Card group 7; soil parameters

159. The data cards in this group define the blast parameter for sand, sandy loam, loam, clay, and rock. The cards must be supplied in that order. Artificial data values which are not zero must be provided for soil types even though the soils are not present in the fortification. However, when the parameters for the last soil present in the fortification have been given, parameters for the remaining soils need not be. For example, if only sand and loam were present in the fortification, required cards would be data for sand, fake data for sandy loam, and data for loam. No data would be required for clay and rock. Note that only one data item is recorded in each card so that additional data items may in the future be added to take advantage of new blast models. Indicate the end of this group by placing a blank card after the last data card.

Group 7 Format

Columns	Format	Units	Description
11-20	F10.0	in.	Blast propagation factor k (see paragraphs 78-81)

Card group 8; structural parameters

160. The structure may be made up of various elements such as posts, beams, slabs, and columns which may in turn be shielded by sandbags. Some may be sized differently, while some may be the same size. Therefore, the data for each component (see card group 3) include a specification which selects the proper structural parameters from the

data in this group. For example, if the structural parameter number is 4, then the data in the fourth card in this group will be used. This feature allows several different components to have the same structural properties. The maximum number of different structural parameter data sets (i.e., cards) is 15. Place a blank card after the last data card to indicate the end of the group.

Group 8 Format

Columns	Format	Units	Description
1-10	F10.0	$in.^2$	Area of the outside face of the cross-sectional area of wooden members
11-20	F10.0	in.	Thickness for concrete; length for wood
21-30	F10.0	in.-lb	Maximum bending moment for wood
31-40	F10.0	seconds	Natural period of vibration for wood
41-50	F10.0	$in.^3$	Volume of the component

Card group 9; failure tree

161. The cards in this group contain the data for the pairs of values that establish the failure tree (see paragraph 92). The first value of the pair is the level of the component in the tree. The second value is the number of the component. The values are recorded in 5 column fields with up to 8 pairs on each card. The maximum number of pairs of values is 200 (i.e., 400 values). The end of the failure tree data is indicated by a blank card.

Group 9 Format

Columns	Format	Units	Description
1-5	I5	none	Tree level of components in the next field
6-10	I5	none	Number of the component at level given in the previous field
11-15	I5	none	Tree level of component in the next field
16-20	I5	none	Number of the component at the level given in the previou field

Continue recording data in the above manner until the tree structure has been defined.

Card group 10;
number of $P_{k/h}$ curves

162. There is only one card in this group and only two data items on that card. The first item controls the optional printed output of the projectile fragmentation data that were supplied in the input. The output is included when the print control variable value is 2. The second item specifies the number (maximum of 50) of $P_{k/h}$ curves that will subsequently be defined with card groups 12, 13, and 14. These curves, for equipment protected by the structure, are described in paragraphs 124-131.

Group 10 Format

Columns	Format	Units	Description
1-5	I5	none	Print control variable (print projectile fragment distribution when the value is 2)
6-10	I5	none	Number of $P_{k/h}$ curves

Card group 11; personnel
$P_{k/h}$ equation coefficients

163. This card group contains the coefficients (of the Kokinakis type) for the equations used to evaluate $P_{k/h}$ for personnel (see paragraphs 122 and 123). The cards must be in the following order with respect to the data therein: head/neck, thorax, abdomen, pelvis, arms, and legs. The coefficients (C1 through C7) are recorded in the first 7 fields. The eighth field is used to identify the card as to data type. The identification may be alphabetic and/or numeric such as HEAD/NK or ARMS 1/2. Place a blank card after the end of the data to indicate the end of the group.

Group 11 Format

Columns	Format	Units	Description
1-10	F10.0	N/A	Coefficient C_1
11-20	F10.0	N/A	Coefficient C_2
21-30	F10.0	N/A	Coefficient C_3
31-40	F10.0	N/A	Coefficient C_4

(Continued)

Group 11 Format

Columns	Format	Units	Description
41-50	F10.0	N/A	Coefficient C_5
51-60	F10.0	N/A	Coefficient C_6
61-70	F10.0	N/A	Coefficient C_7
73-80	A8	none	Label for the data on the card

Card group 12; residual velocity

164. The data in this group are the three sets of Thor coefficients (T_1 through T_5) which are used to calculate the remaining (residual) velocity after passing through water (first card), an arm (second card), or a leg (last card). The label in the last field is used to identify the data in the card (e.g., ARM). Place a blank card after the last data card to indicate the end of this group.

Group 12 Format

Columns	Format	Units	Description
1-5	F5.3	N/A	Thor coefficient T_1
6-10	F5.3	N/A	Thor coefficient T_2
11-15	F5.3	N/A	Thor coefficient T_3
16-20	F5.3	N/A	Thor coefficient T_4
21-25	F5.3	N/A	Thor coefficient T_5
72-80	A8	N/A	Label for the data on the card

Probability of kill curves

165. As mentioned previously (see paragraph 122), the probability of kill after being hit $P_{k/h}$ for protected components (other than personnel) is defined by up to 16 curves, 1 curve for each fragment weight. The curves are defined by a series of points whose coordinates are velocity versus $P_{k/h}$. In addition, each curve has an associated minimum velocity value below which $P_{k/h}$ is assumed to be zero. The family of curves may be thought of as being drawn on a single graph, the graph containing a velocity versus P_k curve for each fragment weight. Since there may be several different protected components, there may also be several different graphs. The user must supply the data for the curves

on each of the graphs via specifications outlined for card groups 13, 14, and 15 for the number of curves specified in card group 10.

Card group 13;
minimum velocity levels

166. This card, or cards in the case of more than 16 curves, contains the minimum velocity levels for each curve. One set of group 13 cards is required.

Group 13 format

167. Record the minimum velocity values for each curve in the 5-column fields starting with columns 1-5, then 6-10, and on out to 76-80. When necessary, continue on the following cards until all values have been recorded. The units are feet per second and the format is 16 I5. The maximum number of velocity minima is 50 (i.e., 1 for each curve).

Card group 14;
weight curves per component

168. The data in this group specify the number of curves used to describe velocity versus $P_{k/h}$ for a component. This is followed by the fragment weight value for each curve, which is in turn followed by values for the number of points on each curve. Note this number of curves specification is different from the number of curves specification in card group 10. Here, the specification is for curves per component, whereas the group 10 specification is for the total number of curves. Also, note that the sum of the curves per component must equal the total number of curves specification. This card group and the associated group 15 cards comprise a group pair that defines $P_{k/h}$ for the one component. The group pairs are repeated to define $P_{k/h}$ for the other components.

Group 14 Format

Columns	Format	Units	Description
1-4	I4	none	Number N of curves used to describe $P_{k/h}$ for this component
5-8	I4	grains	Weight associated with the first curve

(Continued)

Group 14 Format

Columns	Format	Units	Description
9-12	I4	grains	Weight associated with the second curve. Note that a second curve is not required but is usually provided. Continue with weight specifications until N weight values have been recorded. Then begin the number of points specifications
4N+5 through 4N+8			Number of points in the first curve
4N+9 through 4N+12			Number of points in the second curve. Continue the number of point specifications until a value has been given for each of the N curves

Card group 15;
points on the $P_{k/h}$ curves

169. The data in this group consist of the velocity and $P_{k/h}$ coordinates of points for each of the N curves specified in the preceding group 14 card. The point data are recorded as coordinate pairs, velocity first, then $P_{k/h}$. After the points for one curve have been recorded, the coordinate data for the second and subsequent curves are recorded starting in the next available position of the card used for the previous curves data or a new card if the current card is full. Should one curve have more than eight points, the remaining data are recorded on the next card. As mentioned previously, repeat the card group pairs (14 and 15) until velocity versus $P_{k/h}$ data for all of the protected components have been defined. Note that the recorded $P_{k/h}$ value is the actual $P_{k/h}$ multiplied by 100. Thus, the actual $P_{k/h}$ of 0.63 is recorded as 63 in the 3-column field. The last 8 columns of each card is used for labeling the data in the card. The label is printed in the data verification output of the program.

Group 15 Format

Columns	Format	Units	Description
1-4	I4	ft/sec	Velocity value at a point
5-7	I3	$P_{k/h} \times 100$	$P_{k/h}$ at the same (above) point

(Continued)

Group 15 Format

Columns	Format	Units	Description
8-8	1x	none	Blank
9-12	I4	ft/sec	Velocity value at the next point
13-15	I3	$P_{k/h} \times 100$	$P_{k/h}$ at the same (above) point
16	1x	none	Blank
etc.			Continue as above (note the blank column following each $P_{k/h}$) until data for eight points have been recorded on the card or all the curve points for the component have been recorded. The last allowable positions in the card for velocity and $P_{k/h}$ are
57-60	I4	ft/sec	Velocity
61-63	I3	$P_{k/h}/100$	$P_{k/h}$
64-71	9x	none	Blanks
72-80	A8	none	Label for the data in the card

Card group 16; number of fragment zones and weights

170. There is only one card in this group. It contains the specification for the number of the nose bands (see paragraph 35). The maximum value is 40. This group also contains the specification for the number of different weight classes of fragments that will be projected within each nose band. The maximum number of weight classes is 6.

Group 16 Format

Columns	Format	Units	Description
1-5	I5	none	Number of nose bands
6-10	I5	none	Number of fragment weight classes for each nose band

Card group 17; weapon projectile identification

171. The data in this single card are used to identify the weapon/projectile. The card provides for up to 60 alphabetic and/or numeric character positions for this purpose. The program saves these data and subsequently prints them as weapon identification information in the program output.

Group 17 Format

Columns	Format	Units	Description
1-60	A60	none	Weapon/projectile name

Card group 18;
limits of the nose bands

172. The cards in this group contain the nose angles which define the limits of each nose band (see paragraph 40). The angles are recorded in ascending order starting with the upper limit of the first nose band (the lower limit of the first band is assumed to be 0 degrees) and continuing until the upper limit of the last band has been given. Under this procedure, the lower limit of nose band J is taken as the upper limit of band J-1 (the upper limit of the nonexistent band zero is assumed to be 0 degrees). The maximum number of bands (given in the group 16 card) is 40. Note that two full and one half-full data cards are required to record all 40 nose angles. However, the user needs to supply only the number of cards necessary to specify the nose angles for the number of nose bands specified in the group 16 card.

Group 18 Format

Columns	Format	Units	Description
1-5	F5.1	degrees	First nose angle
6-10	F5.1	degrees	Second nose angle
11-15	F5.1	degrees	Third nose angle
			Continue until all values have been given or the card is filled (i.e., 16 values). Record the remaining values on additional cards

Card group 19; number of
fragments in each weight class

173. This card group specifies the number of fragments of each weight class that are contained in the limits of each nose band. Each card will specify the number of fragments in each weight class for a specific nose band. Since there will be one card for each nose band, the cards must be supplied in ascending order by nose band. Note that the group 21 cards contain the weight values associated with the number of fragment values specified in this group. In both this and the group

21 cards, each card contains data for one nose band

Group 19 Format

Columns	Format	Units	Description
1-5	F5.1	none	Number of fragments in weight class 1
6-10	F5.1	none	Number of fragments in weight class 2
11-15	F5.1	none	Number of fragments in weight class 3
16-20	F5.1	none	Number of fragments in weight class 4
21-25	F5.1	none	Number of fragments in weight class 5
26-30	F5.1	none	Number of fragments in weight class 6

Card group 20; fragment velocity in each nose band

174. The data in this group specify one velocity value for each nose band. Consequently, all fragments in a given nose band will have the same initial velocity. The data are recorded on one or two cards in the same order as the nose band limits. The number of values supplied must be the same as the number of nose bands specified in the group 16 card.

Group 20 Format

Columns	Format	Units	Description
1-4	F4.0	ft/sec	Fragment velocity for nose band 1 (21)*
5-8	F4.0	ft/sec	Fragment velocity for nose band 2 (22)*
9-12	F4.0	ft/sec	Fragment velocity for nose band 3 (23)*

* Place values for nose bands 21 through 40 on a second card using the same format as for the first 20 values. A second card is not required unless there are more than 20 nose bands.

Card group 21; fragment weights per nose band

175. As mentioned previously, up to six weight classes are assigned to each nose band. This group establishes the weight values for each class for each of the nose bands. Thus, one card will contain from one to six weights for fragments in one nose band. Consequently, there must be one card for each nose band. The cards are supplied in the same order as the nose band limits.

176. Recall that each card is paired with a corresponding group

19 card. Together, they define the number of fragments of each weight class that are present in a specific nose band. The velocity of each of these fragments is specified by a single entry in the group 20 data.

Group 21 Format

Columns	Format	Units	Description
1-5	F5.0	grains	Fragment weight for weight class 1
6-10	F5.0	grains	Fragment weight for weight class 2
11-15	F5.0	grains	Fragment weight for weight class 3
16-20	F5.0	grains	Fragment weight for weight class 4
21-25	F5.0	grains	Fragment weight for weight class 5
26-30	F5.0	grains	Fragment weight for weight class 6

APPENDIX A: PROGRAM LIMITATIONS AND NEEDED IMPROVEMENTS

Improvements, Where and When

Purpose

1. This appendix points to areas in this program that need improvements. It may also prove useful to future developers of hardness evaluating programs by pointing out where unwarranted shortcuts were made and where the state of the art was inadequate for the task. With this insight and the known capabilities of the existing program, the future developers may build on the progress of this effort.

Where to begin?

2. Computer programs can always be improved. This program is no exception. The questions before us are which areas are most needing improvement and given the fact of limited resources where may the greatest benefits result for a given amount of improvement effort?

Improvement priority

3. Setting priority for areas needing improvement is premature at this stage in the development process. Too many questions such as the extent of available resources, success in parallel developments, user satisfaction, etc., must be resolved before we can say what is most important.

4. For example, the ground shock induced by a near-miss burst can defeat a structure, but airblast-induced ground shock is not modeled in this program. However, correctly modeling ground shock is today a state-of-the-art process involving huge computer codes and hours of computer time. On the one hand, we might decide that accurate ground shock modeling is a most important need, yet be confronted with the reality that accurate modeling is too costly today. We might, on the other hand, decide to develop a crude shock model or simply delay any improvement until parallel efforts in ground shock modeling show promise of meeting this need. For these and other reasons, this appendix does not recommend priority or identify greatest need.

Current Limitations

Exit velocity

5. The exit velocity calculation for all nonhuman materials uses an approximation based on penetration into massive (very thick) sections. This ignores the fact that a projectile of a given weight and moving at a specific velocity will perforate a section that is thicker than the maximum penetration of the projectile into a massive section of the same material.

Structural component modeling

6. The bending moment calculation for wood is probably fairly good, but it is systematically low. It may be a good idea to use 1/7 instead of 1/8 of the calculated moment $F\ell$ as illustrated in Figure 14 of the main text (see also paragraph 87).

7. Steel beams could be treated as wood beams, with special values for bending moment and pulse period. Steel beams will not actually fail in many instances even though inelastically deformed. Thus, a high value for bending moment will allow for the inelastic deformation. The anisotropic strength and unusual geometry of other steel shapes may make their treatment difficult.

8. Multiple elements acting dependently are described as an equivalent independent system. For example, a roof, made of five crossed layers of planks, nailed and glued together, must be treated in a nonstandard fashion. The geometric description includes all layers, but only the bottom two layers of planks are declared critical and they all share two force accumulators. Special values for the critical moment and the natural period are then entered for these layers. The strength of all is given to only a few.

Airblast

9. The pressure-distance equation used for airblast needs to be improved by adding a pulse duration versus distance relationship. Also, the airblast pressure needs to be coupled to the soil.

Shock-induced damage

10. Shock-induced damage is not modeled for the contents of the

structure. However, such damage can readily disable the equipment within a fortification. A great deal of developmental work, both experimental and analytical, needs to be done before such effects can be even grossly modeled.

Delay fuses and fragments

11. In this program, delay fuses are only effective when the projectile enters soil. Also, fragments produced by contact-fused projectiles are ignored.

Proximity fuse modeling

12. The proximity fuse burst is initiated when the distance along the entry path from the projectile to a component is less than the fuse distance. Consequently, the projectile will not burst under near-miss conditions, even though the near-miss is within the fuse distance. This can be overcome by changing the distance measurement from along the entry path to that of the nearest component (i.e., a spherical fuse surface).

Bursts inside the fortification

13. The program's logic assumes that any burst inside the structure will destroy all components and personnel therein. However, the program considers an inside burst as being one caused by a projectile that has previously passed through a structural component. Thus, a projectile which passes through an opening or J-hooks up into the inside will not be identified as an inside burst. This difficulty can be overcome by placing a very weak (e.g., soap film) structural component over all openings and the earth floor.

Penetration models

14. The penetration model used for concrete appears to be valid. The soil and wood penetration models could be improved. All models would benefit from reliable data on standard fragments. Note that the fragments are assumed to impact the targets with the small end and have zero yaw with respect to the long axis. This results in maximum penetration and damage. More realistic damage modeling might result from a random selection of the yaw angle and hence area of the impact face. Here again, additional experimental data are needed.

Horizontal shooting

15. The program will not correctly function if the line of projectile delivery is at or below horizontal. Further, lines only slightly above horizontal cause the program to begin the entry path much too far from the fortification. Straight-on or upward-directed (low gun to high fortification) shots can be simulated by removing the entry path restriction.

Blast vulnerability models

16. The models used to relate vulnerability of structural elements to blast intensity are very crude. More sophisticated models are available, but they need to be tuned to the geometry description used by this program.

Restart capability

17. The program must run from beginning to end (i.e., all attacks) before it produces the final answer, but it does not have any restart provision. Since the program can run a long time (more than an hour on complex problems), this limitation places an unrealistically high reliability requirement on the computer system. Consequently, when attempting to solve large problems, users may experience considerable difficulty and incur high time and money costs due to computer failure. Restart capability would resolve these difficulties as well as make add-on (afterthought) runs practical (e.g., "Let's run 20 more attacks and see what happens.").

Spall and scab fragments

18. Secondary fragments such as those due to spall and scab are not modeled. Thus, the program subjects the protected components and personnel to threat intensity that is less than actually encountered. Adding this refinement would likely involve a probabilistic trace of the scab fragments. Little appears to be gained by modeling spall fragments. The spall damage on the component hit is currently modeled by changing the specific density.

Conventional weapons

19. The objective of this effort was to model fortification vulnerability to nonnuclear weapons. Thus, the program may be considered

as being limited in that it is restricted to conventional weapons. A high level of effort would be required to model nuclear weapons; only the geometry and probability of survival logic would be useful.

Research tool, not user-oriented

20. This program is a research tool. It is not user-oriented in that it does not defend itself against bad data, its card layout is poor, it does not help the user find mistakes, and its output requires interpretation to obtain valid results. In short, users must prepare perfectly correct data; if they do not, bad results may be produced without any indication that they are wrong. Only through experience with the program and careful testing of the input and the results can the researcher feel confident in using this program. Therefore, before the program is released for general use, the input and output sections should be revised to provide the aforementioned quality assurance features. Also, a "targeteers manual" should be published. The manual should tell how to prepare data and interpret results. It should illustrate the program's application to several field fortifications.

Discontinuous $P_{k/h}$ curves

21. The curves for $P_{k/h}$ are modeled as being linear between the stated data points (fragment velocity versus $P_{k/h}$). This introduces abrupt changes in the function form for velocities on either side of the points, which is due to the abrupt (discontinuous) change in slope at each point. A smooth curve representing $P_{k/h}$ between data points could be obtained using existing cubic spline fitting and interpolating subprograms. These subprograms might also be used to improve the interpolation between weights for each curve.

Ricochet

22. Only one material, concrete, is modeled for ricochet. Other models are available for wood, soil, metal, etc. Components are assumed to be undamaged by the fragments which ricochet from their surface. Actually, the damage (i.e., material removed) is greater for some angles of ricochet than that caused by normal impact paths.

Blast-fragment interaction

23. The fragment damage portion of this program is essentially an

afterthought in that it was incorporated well after the major program logic was coded and tested. Perhaps, consequently, the interaction between blast- and fragment-induced damage is solely through the survival probabilities for blast and fragments. Further, the failure tree relates only to the structural components; consequently, redundancy in protected components cannot be modeled. Also, damage done by fragments to structural components is not modeled in terms of its degradation of component strength.

AD-A088 420 ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG--ETC F/G 19/4
DEVELOPMENT OF A PROGRAM FOR COMPUTER-BASED RESISTANCE ANALYSIS--ETC(U)
JUL 80 J B CHEEK

UNCLASSIFIED WES/TR/SL-80-5 NL

2 OF 2
AD A088420

END
DATE FILMED
9-80
DTIC

In accordance with letter from DAEN-RDC, DAEN-ASI dated 22 July 1977, Subject: Facsimile Catalog Cards for Laboratory Technical Publications, a facsimile catalog card in Library of Congress MARC format is reproduced below.

Cheek, James B
Development of a program for Computer-Based Resistance Analysis (COBRA) / by James B. Cheek, Jr. Vicksburg, Miss. : U. S. Waterways Experiment Station ; Springfield, Va. : available from National Technical Information Service, 1980.
86, 6 p. : ill. ; 27 cm. (Technical report - U. S. Army Engineer Waterways Experiment Station ; SL-80-5)
Prepared for Office, Chief of Engineers, U. S. Army, Washington, D. C., under Project No. 4A762719AT40, Task A1, Work Unit 003, and Task A4, Work Unit 001.

1. COBRA (Computer program). 2. Conventional weapons. 3. Hardened installations. 4. Projectiles. 5. Protective structures. 6. Random processes. 7. Structural analysis. 8. Weapons effects. I. United States. Army. Corps of Engineers. II. Series: United States. Waterways Experiment Station, Vicksburg, Miss. Technical report ; SL-80-5.
TA7.W34 no.SL-80-5